國立陽明交通大學
智慧與綠能產學研究所
碩士論文

Institute of Smart Industry and Green Energy
National Yang Ming Chiao Tung University
Master Thesis

論文題目

Partial Compositional Zero-Shot Learning by Uncertainty-Accuracy Mapping and Hierarchical Mixture of Experts

研究生：周峻緯 (Jhou, Jyun-Wei)
指導教授：馬清文 (Ma, Ching-Wen)

中華民國 一一三年二月
February 2024

論文題目

Partial Compositional Zero-Shot Learning by Uncertainty-Accuracy Mapping and Hierarchical Mixture of Experts

研 究 生：周峻緯
指導教授：馬清文

Student: Jhou, Jyun-Wei
Advisor: Dr. Ma, Ching-Wen

國立陽明交通大學
智慧與綠能產學研究所
碩士論文

A Thesis
Submitted to Institute of Smart Industry and Green Energy
College of Artificial Intelligence and Green Energy
National Yang Ming Chiao Tung University
in partial Fulfilment of the Requirements
for the Degree of
Master of Science
in
Artificial Intelligence

February 2024

Taiwan, Republic of China

中華民國 一一三年二月

Acknowledgement

I extend my heartfelt gratitude to Professor Ma, whose unwavering support and profound insights were instrumental in the fruition of my research. Prof. Ma's willingness to share his extensive knowledge and expertise has enriched my understanding immensely, and for that, I am profoundly thankful. Equally, I owe a debt of gratitude to my family and friends, whose unwavering support and encouragement have been my pillar of strength. Their faith in me has been a constant source of inspiration, pushing me to strive for excellence. I also cherish the camaraderie and mutual support shared with my lab colleagues, making the research journey enjoyable and rewarding. Thank you to each one of you for being my guiding stars.

論文題目

學生：周峻緯　　指導教授：馬清文 博士

國立陽明交通大學 智慧與綠能產學研究所

摘　　要

隨著機器學習技術在各行各業中的普及，學者和專家不斷尋求提高預測準確性的方法。集成學習 (Ensemble learning) 作爲一種已被證明的策略，能有效地提升模型的預測性能。其核心觀念是多個模型的預測整合後往往能夠超越單一模型的能力。本研究主要關注: 如何正確的整合這些模型的預測結果，尤其是考慮到每一位專家在不同的情境或數據上的表現。

本研究提出一種新的策略：通過對每個模型的不確定性和錯誤率進行映射和分析，製作出不確定性和錯誤率的映射表，透過對每個模型的映射表，我們可以更精確地識別哪些專家在特定情境下更具信心度和準確性，這不僅可以幫助我們選擇和整合最有信心的預測結果，而且還可以進一步提升整體預測效能。

本研究在組合零次學習 (Compositional Zero-Shot Learning) 和集成學習的眞實資料集中進行模擬，旨在驗證我們的方法如何提升在這些資料籍上面的性能。

關鍵詞: 集成學習、不確定性、組合零次學習、專家混合、模型較準。

Partial Compositional Zero-Shot Learning by Uncertainty-Accuracy Mapping and Hierarchical Mixture of Experts

Student: Jhou, Jyun-Wei Advisor: Dr. Ma, Ching-Wen

Institute of Smart Industry and Green Energy
National Yang Ming Chiao Tung University

Abstract

As machine learning technologies proliferate across various industries, scholars and experts continually seek methods to enhance prediction accuracy. Ensemble learning, a proven strategy, can effectively improve the predictive performance of models. The core idea is that the combined predictions of multiple models often surpass the capabilities of a single model. This research primarily focuses on: how to correctly integrate these models' predictions, especially considering the performance of each expert under different scenarios or data.

This study introduces a new strategy: by mapping and analyzing the uncertainty and error rates of each model, a mapping table of uncertainty and error rates is constructed. Through each model's mapping table, we can more precisely identify which experts have greater confidence and accuracy under specific circumstances. This not only assists us in selecting and integrating the most confident predictions but also further enhances the overall predictive efficacy.

This research conducts simulations on real datasets of Compositional Zero-Shot Learning(CZSL) and Ensemble Learning to verify how our method improves performance on these datasets.

Keyword: Ensemble learning, Uncertainty, Compositional Zero-Shot Learning,Mixture of Experts,Model Calibration

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the current field of machine learning research, Compositional Zero-Shot Learning (CZSL) and ensemble learning have emerged as two major research directions. CZSL, built upon the foundation of Zero-Shot Learning (ZSL), advocates for making predictions on classes that have not been trained on. On the other hand, ensemble learning seeks to enhance prediction accuracy and model generalization by integrating predictions from multiple models.

The introduction of ensemble learning strategies stems from their demonstrated improvement in model performance across diverse applications. By combining predictions from a variety of models, biases and risks of overfitting from individual models might be mitigated. More specifically, this strategy allows the errors of different models to complement each other and ensures the consistency and stability of the overall prediction.

The advent of Compositional Zero-Shot Learning (CZSL) epitomizes the need to address continuously evolving data categories. CZSL offers a mechanism enabling models to predict categories they have never been directly trained on, ensuring high adaptability in dynamic data environments. This approach is especially critical for scenarios where data distribution changes rapidly or where samples are sparse.

1.2 Motivation

In recent years, machine learning has achieved remarkable results across multiple domains, fundamentally transforming data analysis strategies and leading to numerous technological breakthroughs. However, as the scope of applications expands, researchers are becoming increasingly

aware of the importance of prediction capability enhancements and model robustness across various scenarios.

Under the current deep learning technological framework, challenges such as overconfidence and overfitting still plague researchers. Especially within ensemble learning strategies, the majority of research focuses on model combination techniques, with relatively limited exploration into the uncertainty and confidence estimation of each model. This could have severe implications in high-risk scenarios, such as autonomous driving and the medical field.

To address these challenges, this research focuses on the Compositional Zero-Shot Learning (CZSL) approach, aiming to achieve object recognition given known attributes. Moreover, we delve into how to more precisely assess model uncertainty and attempt to incorporate this assessment into ensemble learning strategies. The goal is to devise an ensemble approach that not only possesses high predictive performance but also offers detailed estimation of model uncertainty.

1.3 Goal

- **Goal for Ensemble Learning**: This research seeks to deeply explore ensemble learning strategies, combining predictions from multiple deep learning models to enhance the overall predictive and generalization capabilities of the model. By analyzing the uncertainty and error rate of each model, we aim to determine which models exhibit higher confidence and accuracy in specific contexts. The objective is to improve overall performance and minimize the risk of overfitting.

- **Goal for CZSL**: We also investigate strategies for Compositional Zero-Shot Learning. Within the CZSL context, although all attributes and object concepts are known during training, the model does not encounter any descriptions of seen or unseen attributes and objects. Our goal is to accurately predict objects in scenarios where attributes are already known.

1.4 Contribution

1. **Uncertainty Mapping and Performance Improvement**: We propose a strategy that involves mapping the uncertainty and error rates of each model, allowing for a more precise assessment and recognition of an expert's confidence and accuracy in specific contexts. Based on this mapping strategy, we observed performance improvements in the CZSL dataset across different expert scenarios with algorithmic-level differences.

2. **Expansion of Mapping-Based Ensemble Learning Strategy**: Building on the aforementioned uncertainty and error rate mapping strategy, we further expanded our approach by adopting traditional ensemble learning methods. Especially in representative CZSL datasets such as MIT-States[1] and UT-Zappos[2], we made more accurate predictions for objects with known attributes. Experimental results demonstrated significant improvements in our strategy over baseline models in terms of prediction accuracy and F1-scores.

3. **Deep Feature and Attribute Analysis**: We conducted an in-depth analysis of features and attributes to further explore how these factors affect model performance. These analyses not only allowed us to gain a deeper understanding of the data characteristics but also guided us on how to more effectively adjust the model to achieve better performance.

Chapter 2

Related Works

2.1 Ensemble Learning

Ensemble learning is widely recognized as a core research methodology in the field of machine learning. It aims to enhance the overall model's predictive accuracy and stability by combining the predictions of multiple base models. Depending on the combination strategies, ensemble learning can be categorized into several distinct approaches, such as Bagging, Boosting, AdaBoost, Stacked Generalization, and Mixture of Experts [3][4][5][6].

Products of Expert (POE) and Sum of Expert (SOE) are two techniques within the Mixture of Experts strategy of ensemble learning. The central idea of Mixture of Experts is to treat individual models as different "experts" and rely on their expertise for predictions. Under this framework, POE operates by multiplying the predictions of each expert [7], whereas SOE sums up their predictive outputs.

Simple Voting and Weighted Voting are often associated with Bagging and Boosting strategies [8]. Simple Voting determines the final output based on the majority class from all model predictions, a strategy commonly seen in Bagging methods such as Random Forests. Weighted Voting, on the other hand, assigns different weights to each model. These weights are typically based on the model's performance or other relevant criteria and are frequently applied in Boosting-type methods like AdaBoost.

Recently, Aimar et al. [9] proposed a significant improvement by training a trio of experts during the training phase and adjusting this trio with various resampling rates, ultimately aggregating through POE to diminish biases.

The approach proposed in this study is most akin to the Mixture of Experts strategy, especially in terms of how to combine the outputs of multiple expert models. We introduce an

innovative strategy that not only maps the uncertainty and error rate of each model but also incorporates the combination concepts of POE and SOE. The aim of this strategy is to assess the confidence and predictive accuracy of expert models more precisely in specific scenarios.

2.2 Compositional Zero-Shot Learning(CZSL)

Compositional Zero-Shot Learning (CZSL) was first introduced in Recognition by Parts[10]. This approach enables us to recognize objects that have never been seen before. In traditional Zero-Shot Learning (ZSL), models can not see view images of known categories but also descriptions of both known and unknown categories [11]. However, the distinction of CZSL is that it does not provide descriptions of any seen or unseen attribute-object combinations (e.g., black cat, where black is the attribute, and cat is the object), even though all attribute and object concepts are known during the training phase.

I Misra.[12] proposed a method addressing this issue, building classifiers directly for images and texts, while trying to embed both into a consistent feature space. In contrast, the research by Nirat Saini[13] focuses on the disentanglement of visual features, based on a single textual category, and uses these disentangled features for composition during testing. Furthermore, S Kumar [14] proposed a two-stage training strategy: initially training an object localizer, followed by a joint classifier for images and texts. X Li's work [15] emphasizes designing a siamese network, embedding both images and textual descriptions into a shared embedding space. and adopting the STM method to generate virtual samples, thereby enhancing the encoder's prototype recognition capabilities.

Our baseline model is primarily based on Decomposed Fusion with Soft Prompt (DFSP) [16]. The authors constructed modules specifically for text and image feature extraction, and decomposed textual features, merging them with image features for joint training. Building on this literature's framework, we made relevant optimizations and adjustments to delve deep into specific research questions. Our research structure and problem setting further optimized and adjusted based on this foundation. Unlike the traditional problem setting of CZSL, our method

emphasizes predicting and identifying objects, given known attributes.

2.3 Normalized Uncertainty estimation

In the field of machine learning, uncertainty estimation and quantification are regarded as vital research directions for enhancing model reliability and accuracy. Firstly, the quantification of uncertainty is deemed crucial in establishing reliable and trustworthy machine learning systems. For instance, some studies estimate uncertainty in Recurrent Neural Networks (RNNs) through stochastic discrete state transitions. C Wang et al.[17] provide a schematic comparison of three distinct uncertainty models (MC dropout, Bootstrap model, and GMM) in the context of deep learning and traditional machine learning. M Huang et al.[18] introduced a method named ”Uncertainty-Estimation with Normalized Logits (UE-NL),” which explores classification problems from a probabilistic perspective to learn reliable classification and anomaly detection models.

In my research, the primary approach is to compute the Normalized Uncertainty by calculating the logits predicted by each expert, aiming for a more precise understanding and integration of the experts' prediction outcomes.

2.4 Uncertainty Calibration Error (UCE)

The advancements in deep learning techniques for classification tasks have achieved high accuracy, drawing extensive attention and applications of deep learning classifiers in safety-sensitive areas such as autonomous driving [19] and computer-aided diagnosis [20]. However, solely relying on the high precision of deep learning models is insufficient, especially in these high-risk application scenarios.

When we base crucial decisions on the model's predictions, the uncertainty of model predictions becomes an essential factor. We must deeply understand whether the model's predictions may be biased or if anomalies exist when providing data to the model. Decisions made based on uncertain predictions could have severe consequences [21].

In our research, the Uncertainty Calibration Error (UCE) serves as a primary metric, not only for assessing model performance but also for deriving our proposed Uncertainty to error rate table. Given the interplay between a model's uncertainty and actual prediction error, this mapping offers a quantification method that provides a more precise interpretation and correction of model behavior when confronted with uncertain data.

Specifically, the concept of UCE allows us to delve into the model's confidence intervals in finer detail and offers a mechanism to evaluate and calibrate the reliability of model predictions. Coupled with our "Uncertainty to Error Rate" mapping method, we can not only make decisions based on the model's prediction but also provide predictions that are more interpretable and reliable in specific application scenarios, such as medical image diagnosis or autonomous vehicle navigation.

Chapter 3

Method

3.1 Model Overview

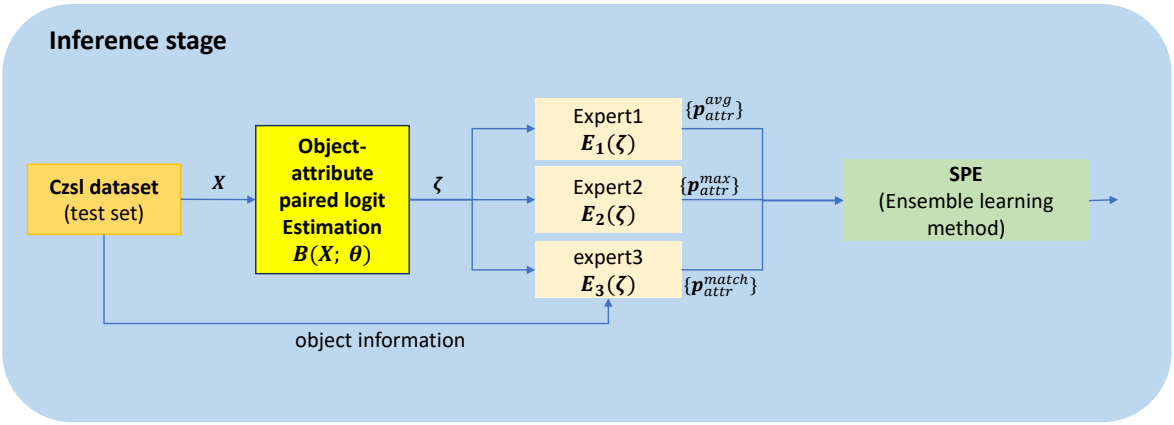

Figure 3.1: Inference by the Object-attribute paired logit Estimation and Multiple Experts

Figure 3.1 illustrates the architecture of our proposed method. Initially, given a set of samples X from the CZSL dataset, we first input them into the Object-attribute paired logit Estimation $B(x;\theta)$, which outputs a pair logit ζ. Then, the pair logit ζ serves as input for Expert 1 $E_1(\zeta)$, Expert 2 $E_2(\zeta)$, and Expert 3 $E_3(\zeta)$, respectively producing the attribute probabilities p_{attr}^{avg}, p_{attr}^{max}, and p_{attr}^{match}, which then serve as inputs for our method's Selected Prime Expert (SPE).

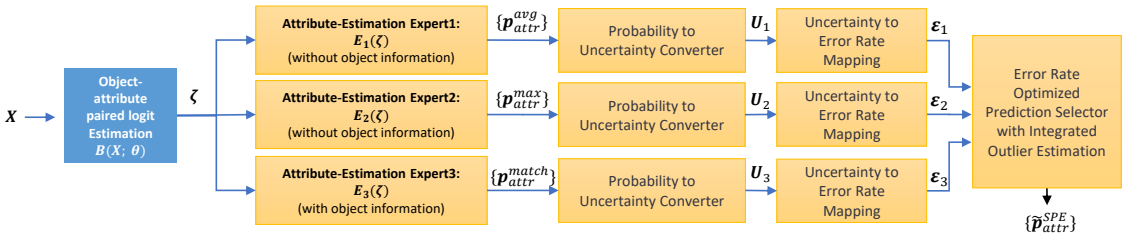

Figure 3.2: Illustration of our method inference stage

3.2 Model Architecture

Figure 3.2 depicts the architecture of our proposed method, Selected Prime Expert (SPE). Given a set of samples X, we initially input them into the Object-attribute paired logit Estimation $B(x;\theta)$, which outputs a pair logit ζ. Subsequently, the pair logit ζ is fed into Expert 1 $E_1(\zeta)$, Expert 2 $E_2(\zeta)$, and Expert 3 $E_3(\zeta)$, each outputting the attribute probabilities p_{attr}^{avg}, p_{attr}^{max}, and p_{attr}^{match}. These probabilities are then converted through a Probability to Uncertainty Converter, resulting in Normalized Uncertainties U_1, U_2, and U_3, which are each processed through Uncertainty to Error Rate Mapping. This mapping transforms the Normalized Uncertainties into error rates ϵ_1, ϵ_2, and ϵ_3 using a conversion table established during the training phase. Finally, these are input into an Error Rate Optimized Prediction Selector with Integrated Outlier Estimation. After excluding outliers, the system selects the best or average probability based on the remaining error rates, ultimately outputting p_{attr}^{SPE}.

The main elements of our framework are:

1. **Object-attribute paired logit Estimation,** $B(x;\theta)$. We employ the Decomposed Fusion with Soft Prompt (DFSP) model as the foundational model for our proposed methods

SPE and HMOE. Its input is X, and it outputs pair logit $\zeta \in \mathcal{R}^{n_{sample} \times n_{pair}}$, where n_{pair} denotes all possible combinations of attributes and objects, with $n_{pair} = n_{att} \times n_{obj}$, n_{att} representing the number of attribute categories in the dataset, and n_{obj} representing the number of object categories.

2. **Expert 1,** $E_1(\zeta)$. We extract the corresponding logit for each attribute from the pair logit ζ, calculate the sum for each attribute, and compute the average value of each attribute logit by dividing the sum by the number of objects, outputting the attribute probability p_{attr}^{avg}.

3. **Expert 2,** $E_2(\zeta)$. We extract the corresponding logit for each attribute from the pair logit ζ, calculate the maximum value for each corresponding attribute, and output the attribute probability p_{attr}^{max}.

4. **Expert 3,** $E_3(\zeta)$. We utilize the true category information of the object, initially extracting the corresponding logit for each attribute, then selecting the corresponding attribute logit based on the object's true category, and finally outputting the attribute probability z_{attr}^{match}.

5. **Probability to Uncertainty Converter**. Our inputs are the corresponding attribute probabilities p_{attr}^{avg}, p_{attr}^{max}, and p_{attr}^{match} from the three experts. We calculate Normalized Uncertainty using the probability distributions of the three experts, with the calculation formula $H(p_i) = \frac{1}{\log N} \sum_{i=1}^{N} - p_i \log p_i$, where N represents the number of samples, eventually outputting Normalized Uncertainty U_1, U_2, and U_3.

$$U_1 = \frac{1}{logN} \sum_{i=1}^{N} - (p_{attr}^{avg})_i log(p_{attr}^{avg})_i. \tag{3.1}$$

$$U_2 = \frac{1}{logN} \sum_{i=1}^{N} - (p_{attr}^{max})_i log(p_{attr}^{max})_i, \tag{3.2}$$

$$U_3 = \frac{1}{logN} \sum_{i=1}^{N} - (p_{attr}^{match})_i log(p_{attr}^{match})_i. \tag{3.3}$$

6. **Uncertainty to Error Rate Mapping**. During the inference stage, we map the Normalized Uncertainty values U_1, U_2, and U_3 from each expert to their corresponding mapped

error rates ϵ_1, ϵ_2, and ϵ_3, based on the uncertainty to error rate mapping table established during the training period.

$$U_1 \rightarrow \epsilon_1, \epsilon_1 \in [0, 1]. \tag{3.4}$$

$$U_2 \rightarrow \epsilon_2, \epsilon_2 \in [0, 1]. \tag{3.5}$$

$$U_3 \rightarrow \epsilon_3, \epsilon_3 \in [0, 1]. \tag{3.6}$$

During the training phase, our method evaluates the Normalized Uncertainty of each model's probability outputs and utilizes it to calculate the Uncertainty Calibration Error (UCE), and establishes an Uncertainty to Error Rate Mapping reliability diagram. This allows for the corresponding error rates to be found during the inference stage, as referenced in Figure 3.3.

Uncertainty to Error Rate Mapping reliability diagram (U to E Mapping reliability diagram)

In the Uncertainty to Error Rate Mapping reliability diagram, we divide the uncertainty of all training samples into ten intervals and calculate the corresponding error rate for each interval. This error rate reflects the model's reliability within that range of uncertainty.

We will use this U to E Mapping reliability diagram during the inference stage. If the model is completely reliable, the diagram will form a 45-degree line, indicating a perfect alignment between uncertainty and error rate. In this diagram, the color represents the quantity of samples. More samples result in a yellow color; conversely, with fewer samples, it turns blue, as seen in Figure 3.3.

7. **Error Rate Optimized Prediction Selector with Integrated Outlier Estimation**. Cal-

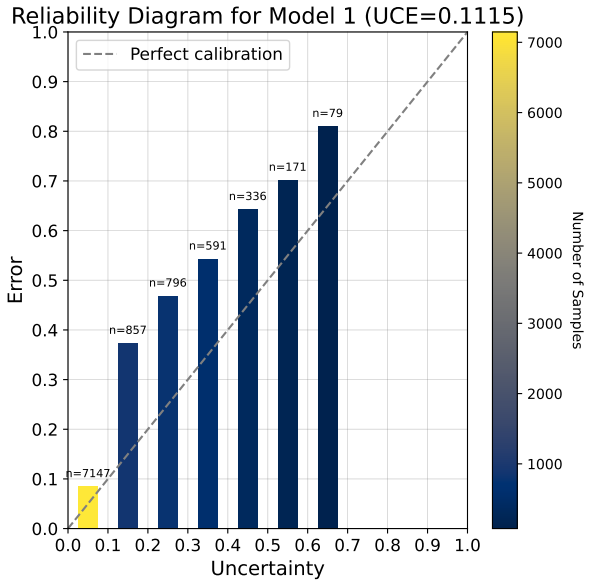

Figure 3.3: U to E Mapping reliability diagram

culate the absolute difference between the error rate of each sample and the minimum error rate. If this difference is less than a defined threshold θ, then the sample is identified as an outlier. For the ith sample, calculate the minimum error rate:

$$\epsilon_{mini} = min(\epsilon_{1i}, \epsilon_{2i}, \epsilon_{3i}) \tag{3.7}$$

Calculate the absolute difference with the minimum error rate: For each sample i and each expert j ($j = 1, 2, 3$), calculate:

$$\Delta\epsilon_{ji} = |\epsilon_{ji} - \epsilon_{mini}| \tag{3.8}$$

Determine the outlier condition: For each sample i and each expert j, if

$$\Delta\epsilon_{j_i} < \theta \tag{3.9}$$

then determine that the sample from this expert is identified as an outlier.

How to Set Thresholds? We will experiment with different threshold combinations on

the validation dataset to find the optimal threshold. Then, this optimal threshold will be applied to the test dataset.

Expert Probability Selection Based on Error Rate Thresholding

P_{j_i}, P_{ki}, and P_{li} are the predicted probabilities for the ith sample from experts j, k, and l, respectively.

$$p_{attr}^{SPE} = \begin{cases} \frac{P_{j_i}+P_{ki}+P_{li}}{3}, & if\, three\, experts\, j,k,l\, satisfy\, \Delta\epsilon_{j_i} < \theta and\, \Delta\epsilon_{ki} < \theta\, and\, \Delta\epsilon_{li} < \theta \\ \frac{P_{j_i}+P_{ki}}{2}, & if\, two\, experts\, j,k\, satisfy\, \Delta\epsilon_{j_i} < \theta\, and\, \Delta\epsilon_{ki} < \theta \\ P_{j_i}, & if\, one\, expert\, j\, satisfies\, \Delta\epsilon_{j_i} < \theta \end{cases} \tag{3.10}$$

In summary, the above method is named: Selected Prime Expert (SPE).

The primary objective of developing the SPE system is to automatically select the best model to ensure optimal performance. By choosing high-performance models, we can avoid a common issue in ensemble learning, where including models with poor performance can decrease the overall performance.

3.3 Our method: Hierarchical MOE(HMOE)

Firstly, we have the model attribute probability outputs from experts 1, 2, and 3, p_{attr}^{avg}, p_{attr}^{max}, and p_{attr}^{match}. We utilize the SPE method to calculate p_{attr}^{SPE}, employ the sum of experts (SOE) method to calculate P_{attr}^{SOE}, and use the product of experts (POE) method to calculate p_{attr}^{POE}, as shown in the equations. (3.11)(3.12)(3.13)。

$$\widehat{p}_{attr}^{SPE} = f(p_{attr}^{avg}, p_{attr}^{max}, p_{attr}^{match}) \tag{3.11}$$

$$\widehat{p}_{attr}^{SOE} = \frac{p_{attr}^{avg} + p_{attr}^{max} + p_{attr}^{match}}{3}. \tag{3.12}$$

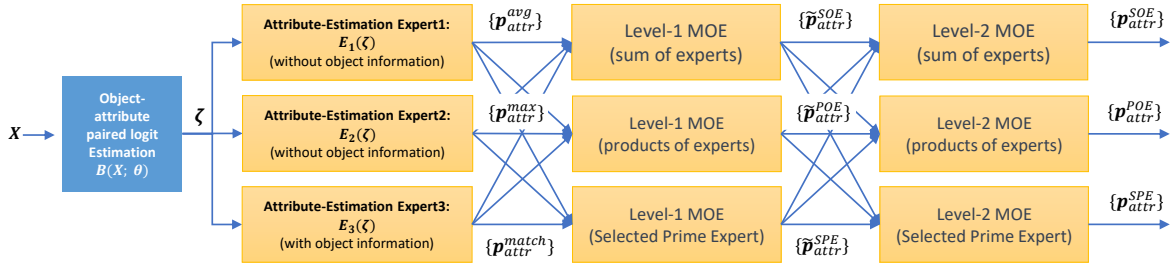

Figure 3.4: Illustration of the Hierarchical MOE(HMOE)

$$\widehat{p}_{attr}^{POE} = (p_{attr}^{avg} \circ p_{attr}^{max} \circ p_{attr}^{match}). \tag{3.13}$$

Lastly, we will again use the SPE method as well as the SOE and POE methods for calculation, ultimately obtaining p_{attr}^{SPE}, p_{attr}^{SOE}, and p_{attr}^{POE}, as illustrated in the equations (3.14), (3.15), and (3.16).

$$p_{attr}^{SPE} = f(\widehat{p}_{attr}^{SPE}, \widehat{p}_{attr}^{POE}, \widehat{p}_{attr}^{SOE}) \tag{3.14}$$

$$p_{attr}^{SOE} = \frac{\widehat{p}_{attr}^{SPE} + \widehat{p}_{attr}^{POE} + \widehat{p}_{attr}^{SOE}}{3}. \tag{3.15}$$

$$p_{attr}^{POE} = (\widehat{p}_{attr}^{SPE} \circ \widehat{p}_{attr}^{POE} \circ \widehat{p}_{attr}^{SOE}). \tag{3.16}$$

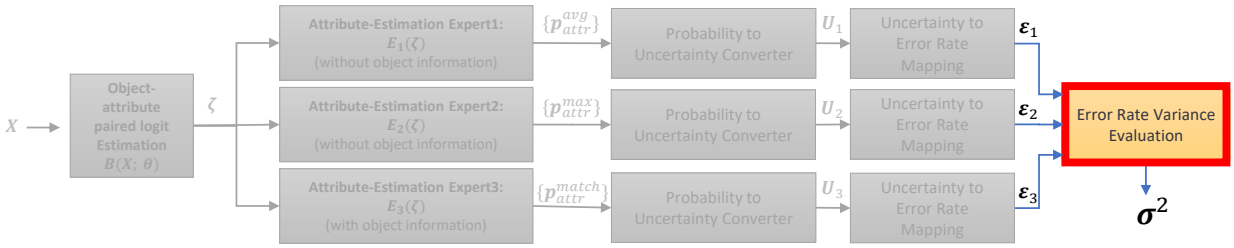

Figure 3.5: Illustration of evaluating the Variance of the Average Error Rate

3.4 Evaluating the Variance of the Average Error Rate

Initially, we have the model error rate outputs from experts 1, 2, and 3, denoted as ϵ_1, ϵ_2, and ϵ_3, as shown in equations (3.4), (3.5), and (3.6). We use ϵ_1, ϵ_2, and ϵ_3 to calculate the variance of the error rates for each sample from these three experts, as illustrated in equation (3.17).

$$\sigma_i^2 = \frac{1}{3}((\epsilon_{1_i} - \bar{\epsilon}_i) + \epsilon_{2_i} - \bar{\epsilon}_i) + \epsilon_{3_i} - \bar{\epsilon}_i)), \bar{\epsilon}_i = \frac{(\epsilon_{1_i} + \epsilon_{2_i} + \epsilon_{3_i})}{3} \tag{3.17}$$

We can average the variance of the error rates for each sample, denoted as σ_{avg}^2, where n represents the number of samples.

$$\sigma_{avg}^2 = \frac{1}{n}\sum_{n}^{i=1}\sigma_i^2 \tag{3.18}$$

3.5 Object-attribute paired logit Estimation Architecture

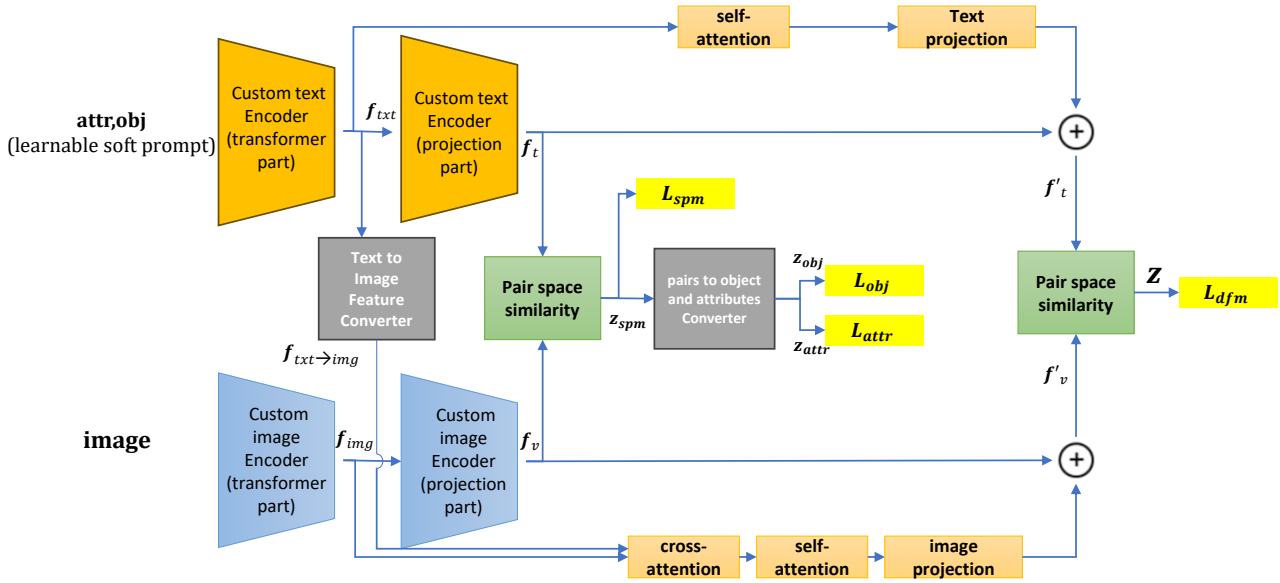

Figure 3.6: Illustration of Object-attribute paired logit Estimation Architecture

Figure 3.6 illustrates the training architecture diagram of our Object-attribute paired logit Estimation using the Decomposed Fusion with Soft Prompt (DFSP) method. The Encoder utilizes the pre-trained model from Contrastive Language-Image Pre-Training (CLIP) [22], which has been pre-trained on approximately 400 million text-image pairs.

The primary components of our framework are:

1. **Custom text Encoder(Transformer part)**. The input consists of text with soft prompts, which are embedded into an embedding space, resulting in the output f_{txt}. The reason for employing learnable soft prompts is that they can replace the fixed token embeddings of clip embeddings, making them learnable.

2. **Custom Image Encoder (Transformer Part)**. The input variable f_{txt} undergoes a linear projection to obtain f_t. This transformation is designed to ensure that f_t has the same

dimensionality as f_v, enabling the computation of Pair Space Similarity.

3. **Custom text Encoder (projection part)**. The input is an image, and its architecture is based on the vision transformer (vit) [23]. It embeds the image into a feature space, producing the output f_{img}.

4. **Custom image Encoder (projection part)**. The input variable f_{img} undergoes a linear projection transformation, resulting in f_v. This transformation is designed so that f_t has the same dimension as f_v.

5. **Text to Image Feature Converter**. We apply a series of dimension transformations on f_{txt} to make it consistent with f_{img}. This is done to allow simultaneous learning of text and image features during the training of the image encoder. The output is $f_{\text{txt}\rightarrow\text{img}}$.

6. **Pair space similarity**. We compute the dot product between the feature embeddings of images and texts to obtain the model outputs z and z_{spm}, as demonstrated in equations (3.19) and (3.20).

$$z_{spm}(\frac{y=(s,o)}{x;\theta}) = \frac{exp(f_t)\cdot f_v}{\sum_{\bar{o},\bar{s}\in C_s\cup C_u} exp(f_t)\cdot f_v} \tag{3.19}$$

$$z(\frac{y=(s,o)}{x;\theta}) = \frac{exp(f'_t)\cdot f'_v}{\sum_{\bar{o},\bar{s}\in C_s\cup C_u} exp(f'_t)\cdot f'_v} \tag{3.20}$$

7. **Pairs to Object and Attributes Converter**. We decompose z_{spm} into z_{att} and z_{obj}. Where z_{obj} represents the average value of all object features, and z_{spm} denotes the average value of all attribute features. Finally, we compute the cross entropy loss using z_{spm} and z_{obj} as demonstrated in the equation(3.21)(3.22).

$$L_{att} = \frac{-1}{|A|}\sum_{(x,y)\in C_s} log(z_{attr}) \tag{3.21}$$

$$L_{obj} = \frac{-1}{|A|}\sum_{(x,y)\in C_s} log(z_{obj}) \tag{3.22}$$

8. **Text Projection and Image Projection**. The text feature f_{txt} processed through self-attention and the image feature f_{img}, which has been fused with text features via cross-attention, are both subjected to Text Projection and Image Projection, respectively. This ensures their dimension consistency with the text feature f_t and the image feature f_v. Finally, we sum up the two text features and the two image features. The weights of these features can be adjusted by setting the coefficient α.

9. L_{spm} **and** L_{dfm}. We compute the cross entropy loss using the model outputs z_{spm} and z respectively, as depicted in the equations (3.21) and (3.22).

$$L_{spm} = \frac{-1}{|A|} \sum_{(x,y)\in C_s} log(z_{spm}) \tag{3.23}$$

$$L_{dfm} = \frac{-1}{|A|} \sum_{(x,y)\in C_s} log(z) \tag{3.24}$$

Chapter 4

Experiment and Results

4.1 Dataset

1. **Compositional Zero-shot Learning Dataset**

   In our study, we primarily selected two representative datasets: MIT-States and UT-Zappos for in-depth analysis and research. Both are commonly used benchmark datasets in current research.

   MIT-States dataset:

   Originating from MIT's Computer Vision Laboratory, this dataset is primarily employed for studying various states and attributes of objects. It comprises approximately 50,000 images, each meticulously annotated, spanning 245 distinct attributes. Furthermore, these attributes span 16 different object categories, such as "wet road" or "damaged shoe." This design ensures that MIT-States encompasses a multitude of object-state combinations, providing a challenging training and testing environment for models.

   UT-Zappos dataset:

   In contrast to MIT-States, the focus of the UT-Zappos dataset is more specific, predominantly studying types and attributes of shoes. Originating from the University of Texas at Austin, it contains roughly 29,000 shoe images. Each image is labeled with 115 distinct attributes, ranging from heel height to material and color, and so on. Additionally, the dataset defines 14 different shoe types, such as athletic shoes, high heels, etc. This allows researchers to delve deep into the subtle differences between shoes, gaining insights into which attributes and features are most crucial for shoe classification.

Table 4.1: Summary for the czsl Dataset.

	MIT-States	**UT-Zappos**
Image Count	50,000	29,000
Attribute Count	245	115
Object Count	16	14

Pair Counts:

Tables 4.2 and 4.3 respectively display an overview of dataset pair counts in the closed-world and open-world settings.

Table 4.2 presents the distribution of attributes (Attr.), objects (Obj.), and seen (y^S) and unseen (y^u) pair categories within the training, validation, and test sets. Specifically, for the MIT-States dataset, it contains 115 attribute categories, 245 object categories, and 1,262 visible pairs in the training set. The validation set comprises 300 seen pairs and 300 unseen pairs, while the test set contains 400 seen pairs and 400 unseen pairs. On the other hand, the structure of the UT-Zappos dataset includes 16 attribute categories, 12 object categories, and 83 seen pairs in its training set, with its validation and test sets containing 15 and 18 seen and unseen pairs, respectively.

In the open-world setting, Table 4.3 showcases the total pair count (n_{pair}) for each dataset. The MIT-States dataset contains a higher number of 28,175 pairs, while the UT-Zappos dataset contains 196 pairs.

Table 4.2: Summary of Dataset Pair Counts in Closed-World Setting

	Train set			Val set		Test set	
Datasets	Attr.	Obj.	y^S	y^u	y^S	y^u	y^S
MIT-States	115	245	1262	300	300	400	400
UT-Zappos	16	12	83	15	15	18	18

Table 4.3: Summary of Dataset Pair Counts in Open-World Setting

Datasets	n_{pair}
MIT-States	28175
UT-Zappos	196

4.2 Evaluation Metrics

Evaluation metrics play an important role in assessing the performance of machine learning models. The confusion matrix stands as one of the fundamental tools for this purpose. In this section, we will elaborate on various metrics derived from the confusion matrix, such as Precision, Recall, F1 Score, FOR, and FDR.

Confusion Matrix. A confusion matrix is a table used to describe the performance of a classification model on a dataset with known true values. The matrix encompasses four primary entries: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), as illustrated in Figure 4.8.

Confusion Matrix		Actual Values	
		Positive	Negative
Predicted	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Figure 4.1: Confusion Matrix

The following are the evaluation metrics adopted in our study:

1. **Precision**. Precision is a metric that quantifies the number of accurate positive predictions. Its formula is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

Precision reflects the accuracy of positive predictions, as depicted in Equation (4.1).

2. **Recall**. Also known as sensitivity or true positive rate, Recall measures the proportion of actual positives that are correctly identified. Its formula is given by:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

Recall emphasizes the capability of the classification model to identify all relevant instances, as depicted in Equation (4.2).

3. **F1 Score**. The F1 Score is the harmonic mean of Precision and Recall, providing a balance between the two. Its formula is given by:

$$F1Score = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{4.3}$$

The F1 Score is especially useful when dealing with imbalanced datasets, as depicted in Equation (4.3).

4. **FOR (False Omission Rate)**. The False Omission Rate quantifies the likelihood of incorrect negative predictions. Its formula is given by:

$$\text{FOR} = \frac{FN}{FN + TN} \tag{4.4}$$

FOR offers insight into the reliability of negative predictions, as shown in Equation (4.4).

5. **FDR (False Discovery Rate)**. The False Discovery Rate evaluates the proportion of false

identifications in positive predictions. It is defined as:

$$\text{FDR} = \frac{FP}{FP + TP} \tag{4.5}$$

The FDR helps in understanding the rate of false alarms in predictions, as illustrated in Equation (4.5).

4.3 Experiment Setup

We employed the CLIP model that pre-trained on the LAION-400M dataset and utilized the PyTorch framework. Our primary objective was to examine the capabilities of this model in specific tasks. To align with our objectives, we adopted the following hyperparameters and settings:

The main architectural choice was CLIP, a model developed and trained on a large dataset, LAION-400M. For our experiments, the input image dimensions were fixed to 3 channels, a height of 224, and a width of 224, ensuring that the model receives standardized image data.

For the training process, we set it for 20 epochs. An epoch, in our context, signifies a complete forward and backward pass of all the training samples. This implies that our model viewed the entire dataset 20 times during its training phase.

The batch size was consistently set at 128. While smaller batch sizes could potentially lead to better generalization, they might also result in slower training due to fewer samples being processed at once. After evaluating the trade-offs, we deemed a batch size of 128 to be optimal for our experiments.

The learning rate, determining the step size during each iteration towards minimizing the loss function, was initialized at 0.0005. Additionally, the optimizer was set to apply weight decay. Weight decay is a regularization technique, typically introduced in the form of an L2 penalty to the loss function. Lastly, all our experiments were conducted on a NVIDIA Tesla V100 GPU.

Table 4.4: Summary of the experiment setup

Feature	Details
GPU	A single NVIDIA Tesla V100 GPU for each experiment
Framework	PyTorch
Backbone	CLIP (pretrained on LAION-400M)
Image size	3 x 224 x 224 (channel x height x width)
Epoch	20
Batch size	128
Learning rate	Initialized at 0.0005 with L2 weight decay
Optimizer	Adam

4.4 SPE Uncertainty to Error Rates Mapping reliability diagram

Figures 4.2, 4.3, and 4.4 illustrate the reliability diagrams for the Uncertainty to Error Rates Mapping generated by our method SPE on the UT-Zappos dataset.

Figures 4.5, 4.6, and 4.7 show the reliability diagrams for the Uncertainty to Error Rates Mapping produced by our method SPE on the MIT-States dataset.

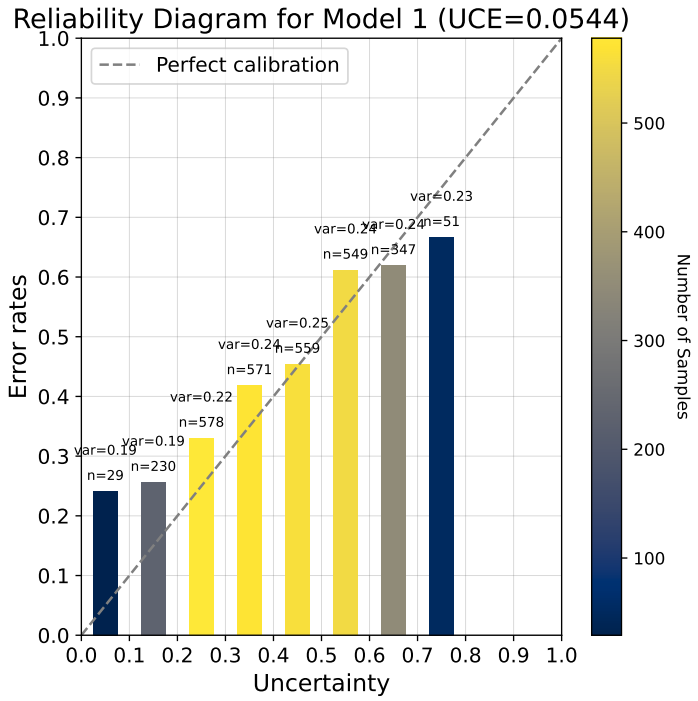

Figure 4.2: U to E Mapping reliability diagram expert1 on UT-Zappos

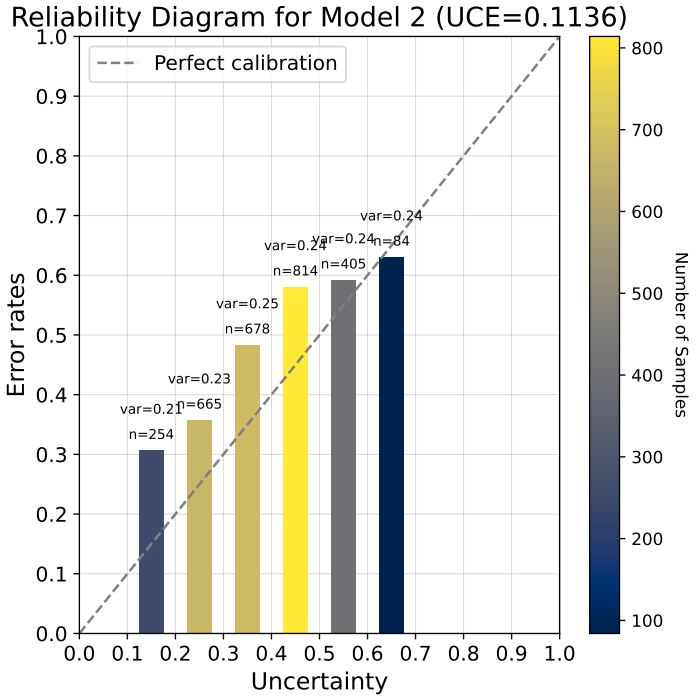

Figure 4.3: U to E Mapping reliability diagram expert2 on UT-Zappos

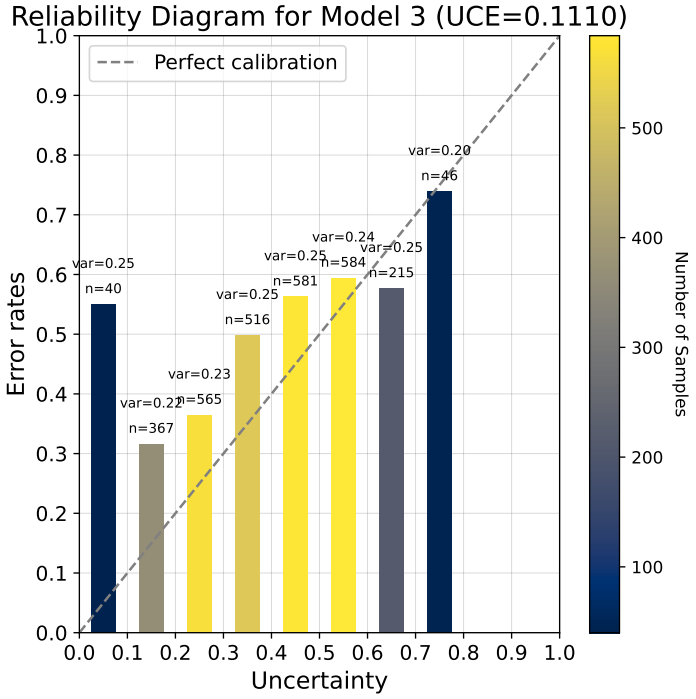

Figure 4.4: U to E Mapping reliability diagram expert3 on UT-Zappos

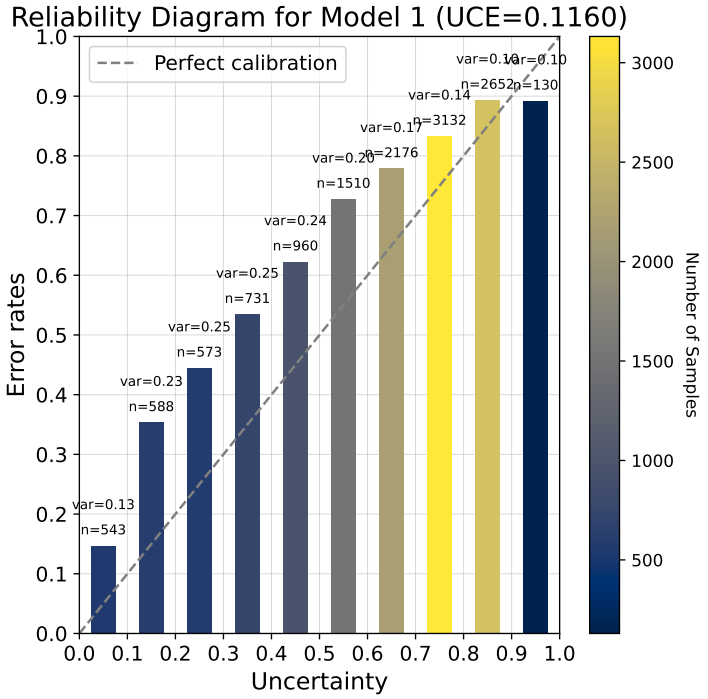

Figure 4.5: U to E Mapping reliability diagram expert1 on MIT-States

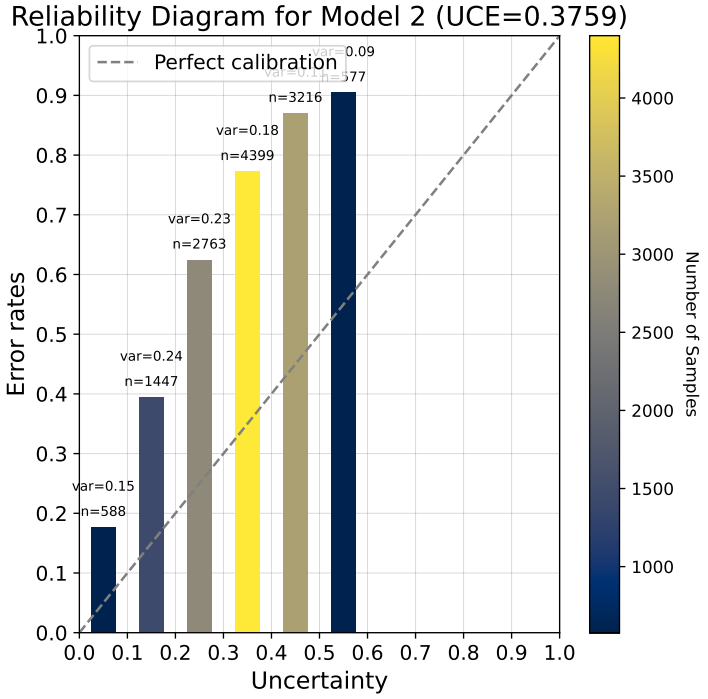

Figure 4.6: U to E Mapping reliability diagram expert2 on MIT-States

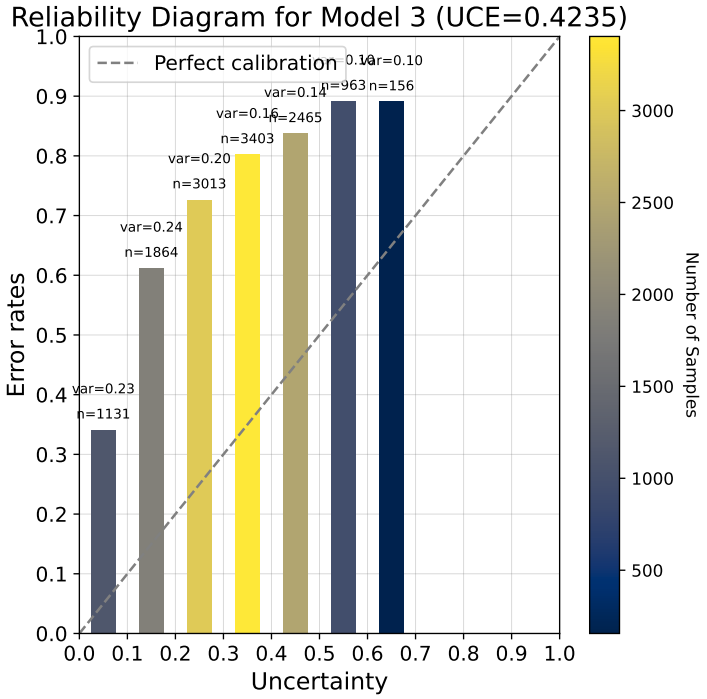

Figure 4.7: U to E Mapping reliability diagram expert3 on MIT-States

4.5 Experiment (I): SPE with Dynamic Thresholding

Our method involves predicting attributes within the known information of objects. The primary objective is to select thresholds based on the error rate to eliminate outliers, in order to study the variation of the SPE (Selected Prime Expert) outlier integration method under different threshold conditions. We will conduct a series of threshold tests for our SPE method, specifically including thresholds of 0.005, 0.001, 0.05, 0.01, and 0.1. These tests will be carried out on two datasets: MIT-States and UT-Zappos. To ensure the stability and reliability of the results, we will use three different random seed values for multiple experiments and average these results to obtain the final experimental data.

From Table 4.5 and Table 4.7, we can see that SPE, with dynamic thresholding, surpasses other ensemble learning methods, where ”ep123” represents the performance of using ensemble learning by expert 1, expert 2, and expert 3. This finding highlights the key role of dynamic threshold settings in enhancing the efficiency of the SPE algorithm, especially in the application context of dealing with UT-Zappos and MIT-States datasets.

By analyzing Table 4.6 and Table 4.8, we can observe that different threshold settings significantly impact model performance. Specifically, when dealing with the UT-Zappos dataset, setting the threshold to 0.01 achieved the best performance. On the MIT-States dataset, setting the threshold to 0.05 could realize better performance. These results highlight the impact of threshold selection on SPE efficiency across different datasets.

Table 4.5: Performance Comparison of SPE with Other Ensemble Learning Methods on the UT-Zappos Dataset

Model Configuration	HMOE-SOE	HMOE-POE	HMOE-SPE	SPE	SOE	POE
ep123 acc(↑)	0.5349 ± 0.003	0.5312 ± 0.001	0.5275 ± 0.001	**0.5372 ± 0.004**	0.5290 ± 0.003	0.5275 ± 0.001

Table 4.6: SPE with Dynamic Thresholding on UT-Zappos Dataset

Threshold	SPE ep123 acc(↑)
0.005	0.5349 ± 0.021
0.001	0.5347 ± 0.02
0.05	0.5352 ± 0.019
0.01	**0.5372 ± 0.025**
0.1	0.5338 ± 0.02

Table 4.7: Performance Comparison of SPE with Other Ensemble Learning Methods on the MIT-States Dataset

Model Configuration	HMOE-SOE	HMOE-POE	HMOE-SPE	SPE	SOE	POE
Ep123 acc(↑)	0.3044 ± 0.019	0.2667 ± 0.009	0.2568 ± 0.013	**0.3110 ± 0.008**	0.2908 ± 0.028	0.2568 ± 0.013

Table 4.8: SPE with Dynamic Thresholding on MIT-States Dataset

Threshold	SPE ep123 acc(↑)
0.005	0.3104 ± 0.008
0.001	0.3106 ± 0.008
0.05	$\mathbf{0.3110 \pm 0.018}$
0.01	0.3098 ± 0.011
0.1	0.3095 ± 0.011

4.6 Experiment (II): Comparison on Other Ensemble Learning Methods

Our method involves predicting attributes within the known information of objects. This study aims to evaluate and compare the performance of our proposed methods (HMOE, SPE) against traditional ensemble learning methods (SOE, POE, Simple voting, Weighted voting) on two different CZSL datasets, using two evaluation metrics (accuracy, F1 score). To ensure the robustness and reproducibility of the experimental results, we will use three different random seed values for multiple experiments and average these results to obtain the final experimental data.

Data analysis from Table 4.9 and Table 4.10 shows that the SPE method exhibits a certain degree of advantage over traditional techniques. However, it is noteworthy that in the evaluation of expert 12, a slight performance improvement is shown when using the HMOE-SOE method relative to SPE, and for the evaluation of expert 13, when applying F1-Score as the evaluation metric, HMOE-SOE also shows certain advantages over SPE. Besides, in most cases, SPE outperforms other methods. This analysis indicates that although SPE performs well in most situations, the HMOE-SOE method may offer better performance under specific conditions.

Data analysis from Table 4.11 and Table 4.12 concludes that the SPE method surpasses other ensemble learning methods in overall performance. Whether using accuracy or F1-Score as the

performance evaluation metric, SPE exhibits significant advantages.

Table 4.9: Comparison of Other Ensemble Learning Methods on the UT-Zappos Dataset using Accuracy

	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE	Simple voting	Weighted voting
ep12 acc (↑)	0.5357 ± 0.004	**0.5376 ± 0.004**	0.5352 ± 0.006	0.5321 ± 0.001	0.5328 ± 0.003	0.5321 ± 0.001	0.5191 ± 0.009	0.5191 ± 0.009
ep13 acc (↑)	**0.5412 ± 0.005**	0.5374 ± 0.001	0.5340 ± 0.002	0.5306 ± 0.0001	0.5321 ± 0.001	0.5306 ± 0.0001	0.5291 ± 0.009	0.5241 ± 0.016
ep23 acc (↑)	**0.5140 ± 0.001**	0.5121 ± 0.0001	0.5129 ± 0.000	0.5139 ± 0.001	0.5138 ± 0.0001	0.5139 ± 0.0001	0.5127 ± 0.0001	0.5127 ± 0.0001
ep123 acc (↑)	**0.5372 ± 0.004**	0.5349 ± 0.003	0.5312 ± 0.001	0.5275 ± 0.001	0.5290 ± 0.003	0.5275 ± 0.001	0.5141 ± 0.002	0.5177 ± 0.007

Table 4.10: Comparison of Other Ensemble Learning Methods on the UT-Zappos Dataset using F1 Score

	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE	Simple voting	Weighted voting
ep12 F1 score (↑)	0.2922 ± 0.003	**0.2974 ± 0.004**	0.2923 ± 0.003	0.2844 ± 0.006	0.2837 ± 0.001	0.2844 ± 0.006	0.2803 ± 0.003	0.2803 ± 0.003
ep13 F1 score (↑)	0.2884 ± 0.017	**0.2957 ± 0.008**	0.2896 ± 0.006	0.2806 ± 0.007	0.2876 ± 0.008	0.2806 ± 0.007	0.2803 ± 0.003	0.2810 ± 0.002
ep23 F1 score (↑)	**0.2941 ± 0.005**	0.2832 ± 0.006	0.2832 ± 0.006	0.2839 ± 0.006	0.2844 ± 0.007	0.2839 ± 0.006	0.2846 ± 0.010	0.2846 ± 0.010
ep123 F1 score (↑)	**0.2977 ± 0.004**	0.2962 ± 0.005	0.2909 ± 0.008	0.2866 ± 0.009	0.2918 ± 0.008	0.2866 ± 0.009	0.2810 ± 0.002	0.2853 ± 0.009

Table 4.11: Comparison of Other Ensemble Learning Methods on the MIT-States Dataset using Accuracy

	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE	Simple voting	Weighted voting
ep12 acc (↑)	**0.3056 ± 0.012**	0.2977 ± 0.027	0.2866 ± 0.029	0.2591 ± 0.060	0.2703 ± 0.048	0.2594 ± 0.059	0.2717 ± 0.014	0.2771 ± 0.006
ep13 acc (↑)	**0.2812 ± 0.018**	0.2750 ± 0.023	0.2588 ± 0.012	0.2331 ± 0.029	0.2660 ± 0.019	0.2543 ± 0.031	0.2758 ± 0.021	0.2758 ± 0.021
ep23 acc (↑)	**0.3113 ± 0.007**	0.3036 ± 0.018	0.2671 ± 0.006	0.2676 ± 0.004	0.2931 ± 0.024	0.2650 ± 0.005	0.2627 ± 0.011	0.2627 ± 0.011
ep123 acc (↑)	**0.3110 ± 0.008**	0.3044 ± 0.019	0.2667 ± 0.009	0.2568 ± 0.013	0.2908 ± 0.028	0.2568 ± 0.013	0.2681 ± 0.010	0.2848 ± 0.013

Table 4.12: Comparison of Other Ensemble Learning Methods on the MIT-States Dataset using F1 Score

	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE	Simple voting	Weighted voting
ep12 F1 score (↑)	**0.2326 ± 0.003**	0.2275 ± 0.011	0.2230 ± 0.024	0.1960 ± 0.045	0.2168 ± 0.023	0.1962 ± 0.045	0.2113 ± 0.018	0.2213 ± 0.002
ep13 F1 score (↑)	**0.2286 ± 0.015**	0.2259 ± 0.017	0.2125 ± 0.010	0.1867 ± 0.028	0.2259 ± 0.016	0.2007 ± 0.036	0.2186 ± 0.007	0.2086 ± 0.017
ep23 F1 score (↑)	**0.2394 ± 0.001**	0.2355 ± 0.008	0.2188 ± 0.005	0.2196 ± 0.005	0.2236 ± 0.005	0.2050 ± 0.015	0.2122 ± 0.022	0.2186 ± 0.007
ep123 F1 score (↑)	**0.2412 ± 0.003**	0.2387 ± 0.010	0.2183 ± 0.009	0.2116 ± 0.011	0.2349 ± 0.016	0.2116 ± 0.011	0.2122 ± 0.022	0.2149 ± 0.022

4.7 Experiment (III): SPE Performance Analysis

In previous studies, we observed a significantly higher variance of the error rate in correctly predicted samples by SPE. This phenomenon implies that our SPE method is more capable of generating accurate predictions when the predicted error rate exhibits high variability. Table 4.13 provides empirical evidence for this observation. As the data indicates, SPE's σ^2_{avg} in correctly predicted samples is significantly higher than that in incorrectly predicted samples, whether on the UT-Zappos or MIT-States datasets.

Table 4.13: Error Rate Variance Comparison between UT-Zappos and MIT-States Datasets

Dataset	**Average error rate variance of SPE (correctly predicted samples)** σ^2_{avg}	**Average error rate variance of SPE (incorrectly predicted samples)** σ^2_{avg}
UT-Zappos	0.0046±0.001	0.0039±0.001
MIT-States	0.0268±0.001	0.0160±0.0008

4.8 Ablation Study (I): SPE Outlier Estimation via POE Implementation

In our research, we replaced the SOE component in the SPE outlier estimation framework with POE, aiming to study the impact of this algorithm replacement on SPE performance. Through this modification, we sought to assess the new algorithm's efficacy in enhancing outlier detection, by conducting a series of experiments to confirm its effectiveness.

We primarily altered the probability ensemble method based on Error Rate Thresholding.

Expert Probability Selection Based on Error Rate Thresholding

Let P_{ji}, P_{ki}, and P_{li} be the predicted probabilities for the ith sample by experts j, k, and l, respectively.

$$p_{attr}^{SPE} = \begin{cases} P_{j_i} \circ P_{ki} \circ P_{li}, & if\ three\ experts\ j,k,l\ satisfy\ \Delta\epsilon_{j_i} < \theta and\ \Delta\epsilon_{ki} < \theta\ and\ \Delta\epsilon_{li} < \theta \\ P_{j_i} \circ P_{ki}, & if\ two\ experts\ j,k\ satisfy\ \Delta\epsilon_{j_i} < \theta\ and\ \Delta\epsilon_{ki} < \theta \\ P_{j_i}, & if\ one\ expert\ j\ satisfies\ \Delta\epsilon_{j_i} < \theta \end{cases} \tag{4.6}$$

From the data analysis in Table 4.14 and Table 4.15, we observed a slight decline in performance with SPE-POE compared to the original SPE framework. This finding indicates that using SOE in SPE is more effective than POE.

Table 4.14: SPE Outlier Estimation via POE Implementation on UT-Zappos

	SPE	SPE-POE	SOE	POE
ep123 acc (↑)	**0.5372 ± 0.004**	0.5249 ± 0.021	0.5290 ± 0.003	0.5275 ± 0.001
ep123 F1 score(↑)	**0.2977 ± 0.004**	0.2934 ± 0.002	0.2918 ± 0.008	0.2866 ± 0.009

Table 4.15: SPE Outlier Estimation via POE Implementation on MIT-States

	SPE	SPE-POE	SOE	POE
ep123 acc (↑)	**0.3110 ± 0.008**	0.3038 ± 0.016	0.2908 ± 0.028	0.2568 ± 0.013
ep123 F1 score(↑)	**0.2412 ± 0.003**	0.2408 ± 0.001	0.2349 ± 0.016	0.2116 ± 0.011

4.9 Ablation study(II):SPE Choose the lowest error rate

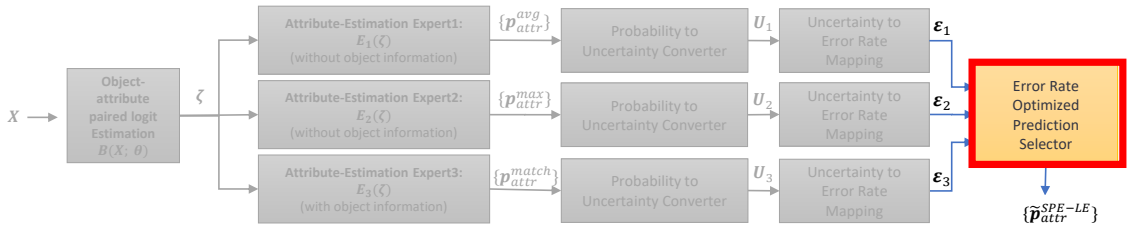

Figure 4.8: Illustration of SPE-LE(SPE-lowest error)

In this study, we delve into the impact of applying different error rate selection strategies on the performance of the SPE method. Specifically, we adopted a selection mechanism based on the lowest error rate, aimed at optimizing the overall performance of the SPE method. Through this approach, we introduced the SPE-LE (SPE Lowest Error) variant, whose core idea is to prioritize those models with the lowest error rates in the decision-making process, thereby ultimately outputting $\widehat{p}_{attr}^{SPE-LE}$.

Error Rate Lookup and Expert Selection: This function selects experts with lower error rates based on the corresponding error rate and uncertainty values for each sample.

Final Prediction Generation: This function selects experts with lower error rates based on the corresponding error rate and uncertainty values for each sample.

$$p_{attr}^{SPE-LE} = \begin{cases} \widehat{y}_1, & if(\epsilon_1 < \epsilon_2) \cap (\epsilon_1 < \epsilon_3) \\ \widehat{y}_2, & if(\epsilon_2 < \epsilon_1) \cap (\epsilon_2 < \epsilon_3) \\ \widehat{y}_3, & if(\epsilon_3 < \epsilon_1) \cap (\epsilon_3 < \epsilon_2) \end{cases} \tag{4.7}$$

From the data analysis in Table 4.16 and Table 4.17, we can observe that SPE exhibits significant advantages in the vast majority of cases. The only exception occurs on the UT-Zappos dataset, where SPE-LE shows a slight advantage when using the F1 score as the evaluation metric. This indicates that, although SPE maintains its superiority across a broad range, under specific conditions, SPE-LE's performance also has the potential to surpass SPE.

Table 4.16: SPE-LE and Other Ensemble Learning Methods on the UT-Zappos Dataset

Dataset: UT-Zappos	SPE	SPE-LE	SOE	POE
ep123 acc (↑)	**0.5372 ± 0.004**	0.5371 ± 0.004	0.5290 ± 0.003	0.5275 ± 0.001
ep123 F1 score(↑)	0.2977 ± 0.004	**0.2988 ± 0.006**	0.2918 ± 0.008	0.2866 ± 0.009

Table 4.17: SPE-LE and Other Ensemble Learning Methods on the MIT-States Dataset

Dataset: MIT-States	SPE	SPE-LE	SOE	POE
ep123 acc (↑)	**0.3110 ± 0.008**	0.3038 ± 0.015	0.2908 ± 0.028	0.2568 ± 0.013
ep123 F1 score(↑)	**0.2412 ± 0.003**	0.2346 ± 0.008	0.2349 ± 0.016	0.2116 ± 0.011

4.10 Ablation study(III):Hierarchical MOE(HMOE)

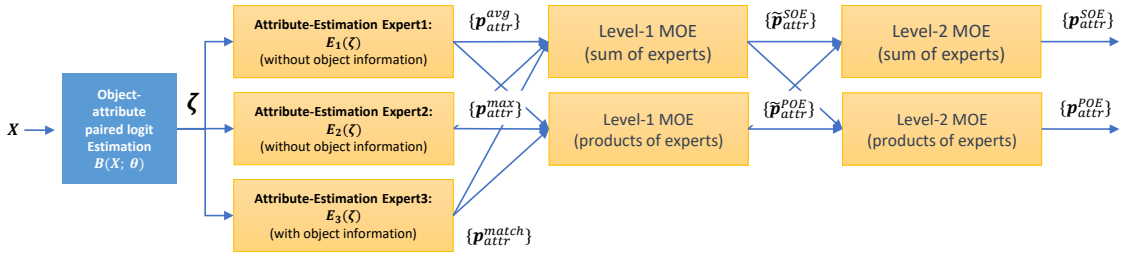

Figure 4.9: Illustration of Hierarchical MOE(HMOE)

Through the removal of the SPE method, this study aims to delve deeper into its specific impact on HMOE performance.

Figure 4.9 shows the architectural diagram of HMOE with SPE removed. This ablation study helps to validate the importance of the SPE method in enhancing or affecting HMOE, providing an important reference basis for subsequent research.

Data analysis from Table 4.18 and Table 4.19 indicates that removing SPE results in a significant decrease in HMOE model performance. This finding suggests that the HMOE model relies on SPE to integrate prediction results, thereby boosting its overall performance.

Table 4.18: The performance Effect of HMOE remove SPE on UT-Zappos

Dataset: UT-Zappos	$\mathbf{p}_{\mathbf{attr}}^{\mathbf{SOE}}$	$\mathbf{p}_{\mathbf{attr}}^{\mathbf{POE}}$	$\mathbf{p}_{\mathbf{attr}}^{\mathbf{SPE}}$	$\mathbf{p}_{\mathbf{attr}}^{\mathbf{SOE}}$ (remove SPE)	$\mathbf{p}_{\mathbf{attr}}^{\mathbf{POE}}$ (remove SPE)
ep123 acc(↑)	**0.5244 ± 0.018**	0.5196 ± 0.017	0.5165 ± 0.016	0.5141 ± 0.018	0.5137 ± 0.013

Table 4.19: The performance Effect of HMOE remove SPE on MIT-States

Dataset: MIT-States	$\mathbf{p}_{\text{attr}}^{\text{SOE}}$	$\mathbf{p}_{\text{attr}}^{\text{POE}}$	$\mathbf{p}_{\text{attr}}^{\text{SPE}}$	$\mathbf{p}_{\text{attr}}^{\text{SOE}}$ (remove SPE)	$\mathbf{p}_{\text{attr}}^{\text{POE}}$ (remove SPE)
ep123 acc(↑)	**0.3038** ± **0.015**	0.2908 ± 0.028	0.2568 ± 0.013	0.2828 ± 0.011	0.2531 ± 0.012

4.11 Ablation study(IV):The Effect of Model Calibration

Considering the overconfidence issue in our model, we can utilize the SPE method to map uncertainties to the error rate lookup table to determine the error rates of each expert. By selecting predictions with the lowest error rates, we can address the issue of model overconfidence.

Our SPE method's performance will be better than that of normally calibrated models, despite the model being overly confident without the need for model calibration to correct it. The following two experiments demonstrate that uncertainty error rate mapping is effective for model calibration compared to other model calibration methods.

1. **Using Monte Carlo Dropout to Calibrate Models**.
2. **Using Temperature scaling to Calibrate Models**.

- **Using Monte Carlo Dropout to Calibrate Models**. In this section, we adopted Monte Carlo Dropout as a technique for model calibration, aiming to explore the extent of its impact on our proposed SPE method after such calibration. To analyze its benefits thoroughly, we selected two representative datasets from the CZSL domain for experimental validation. Moreover, to comprehensively evaluate the efficacy of our method, we introduced a variety of evaluation metrics. Through this series of experiments, we aim to gain a deeper understanding of the specific impact of model calibration on our proposed SPE method.

Based on the analysis conducted on the UT-Zappos dataset, referring to the results shown in Table 4.20 and Table 4.21, we observed that SPE outperforms other methods across both evaluation metrics. Notably, after model calibration, the performance of SPE, SOE, and POE all decreased. Overall, the uncalibrated SPE is capable of outperforming SPE, SOE, and POE that have undergone model calibration.

Similarly, based on the results from the MIT-States dataset, referring to Table 4.22 and Table 4.23, we found that uncalibrated SPE is equally effective in replacing calibrated SPE, SOE, and POE. This result is consistent with the findings on the UT-Zappos dataset, further confirming that SPE can provide considerable performance even without calibration. In summary, these results highlight the importance of selecting appropriate models and calibration strategies for specific datasets. For SPE, its superior performance when uncalibrated indicates that foregoing model calibration is a better choice when selecting and optimizing models.

Table 4.20: Use Monte Carlo dropout for model calibration on UT-Zappos with accuracy

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 acc(↑)	**0.5357 ± 0.004**	0.5328 ± 0.003	0.5321 ± 0.001	0.5062 ± 0.018	0.5061 ± 0.021	0.5055 ± 0.020
Ep13 acc(↑)	**0.5412 ± 0.005**	0.5321 ± 0.001	0.5306 ± 0.000	0.4986 ± 0.019	0.4959 ± 0.016	0.4979 ± 0.021
Ep23 acc(↑)	**0.5140 ± 0.001**	0.5138 ± 0.000	0.5139 ± 0.001	0.5103 ± 0.020	0.5082 ± 0.018	0.5093 ± 0.017
Ep123 acc(↑)	**0.5372 ± 0.004**	0.5290 ± 0.003	0.5275 ± 0.001	0.5062 ± 0.021	0.5060 ± 0.020	0.5041 ± 0.019

Table 4.21: Use Monte Carlo dropout for model calibration on UT-Zappos with F1 score

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 F1 score(↑)	**0.2922 ± 0.003**	0.2837 ± 0.001	0.2844 ± 0.006	0.2791 ± 0.015	0.279 ± 0.012	0.278 ± 0.011
Ep13 F1 score(↑)	**0.2884 ± 0.017**	0.2876 ± 0.008	0.2806 ± 0.007	0.2807 ± 0.003	0.2729 ± 0.007	0.2758 ± 0.012
Ep23 F1 score(↑)	**0.2941 ± 0.005**	0.2844 ± 0.007	0.2839 ± 0.006	0.2778 ± 0.009	0.2788 ± 0.008	0.2792 ± 0.013
Ep123 F1 score(↑)	**0.2977 ± 0.004**	0.2918 ± 0.008	0.2866 ± 0.009	0.2799 ± 0.007	0.2832 ± 0.012	0.2774 ± 0.015

Table 4.22: Use Monte Carlo dropout for model calibration on MIT-States with accuracy

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 acc(↑)	**0.3056 ± 0.012**	0.2703 ± 0.048	0.2594 ± 0.059	0.3024 ± 0.018	0.2995 ± 0.021	0.2890 ± 0.009
Ep13 acc(↑)	**0.2812 ± 0.018**	0.2660 ± 0.019	0.2543 ± 0.031	0.2811 ± 0.008	0.2810 ± 0.022	0.2697 ± 0.013
Ep23 acc(↑)	**0.3113 ± 0.007**	0.2931 ± 0.024	0.2650 ± 0.005	0.3042 ± 0.022	0.2982 ± 0.020	0.2762 ± 0.020
Ep123 acc(↑)	**0.3110 ± 0.008**	0.2908 ± 0.028	0.2568 ± 0.013	0.3108 ± 0.015	0.3102 ± 0.022	0.2647 ± 0.015

Table 4.23: Use Monte Carlo dropout for model calibration on MIT-States with F1 score

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 F1 score(↑)	**0.2326 ± 0.003**	0.2168 ± 0.023	0.1962 ± 0.045	0.2289 ± 0.002	0.2284 ± 0.005	0.2272 ± 0.011
Ep13 F1 score(↑)	0.2286 ± 0.015	0.2259 ± 0.016	0.2007 ± 0.036	0.2278 ± 0.010	0.2272 ± 0.011	**0.223 ± 0.002**
Ep23 F1 score(↑)	**0.2394 ± 0.001**	0.2236 ± 0.005	0.2050 ± 0.015	0.2387 ± 0.003	0.2391 ± 0.003	0.2129 ± 0.012
Ep123 F1 score(↑)	0.2412 ± 0.003	0.2349 ± 0.016	0.2116 ± 0.011	0.2373 ± 0.001	0.2332 ± 0.011	0.2189 ± 0.022

- **Using Temperature scaling to Calibrate Models** .

  In this section of our study, we adopted the Temperature scaling calibration technique to calibrate deep learning models, aiming to investigate the specific impact of this calibration strategy on the SPE method. To delve into its potential benefits, we chose two representative datasets from the CZSL domain as the experimental basis. Furthermore, to comprehensively evaluate the effectiveness of our proposed method, we also incorporated a variety of evaluation metrics. Through this series of experimental designs, we hope to gain a more precise understanding of the impact model calibration has on the SPE method.

  Based on the results from Tables 4.24 and 4.25, some trends can be observed. First, in contrast to the strategy incorporating Monte Carlo Dropout, we find that after introducing

Temperature scaling calibration, there is no significant difference in accuracy, maintaining consistent performance. More importantly, the model's Uncertainty Calibration Error (UCE) showed notable improvement under different calibration temperatures.

Specifically, for the UT-Zappos dataset, when the calibration temperature was set to 1.2, UCE achieved the best performance. Conversely, for the MIT-States dataset, a calibration temperature of 1.4 presented the best UCE. These results indicate that appropriate temperature calibration can make the model's uncertainty predictions more accurate, thus bringing the model's uncertainty closer to the real-world uncertainty.

Table 4.24: The Impact of Temperature scaling on UT-Zappos: Attribute Accuracy and UCE

Model Configuration	ep1 attr_acc(↑)	ep2 attr_acc(↑)	ep3 attr_acc(↑)	ep1 uce(↓)	ep2 uce(↓)	ep3 uce(↓)
UT-Zappos	0.4931 ± 0.012	0.4921 ± 0.013	0.4838 ± 0.015	0.0730 ± 0.001	0.3390 ± 0.002	0.1058 ± 0.002
UT-Zappos Temperature=1.1	0.4931 ± 0.012	0.4921 ± 0.013	0.4838 ± 0.015	0.0474 ± 0.002	0.3036 ± 0.001	0.0665 ± 0.011
UT-Zappos Temperature=1.2	0.4931 ± 0.012	0.4921 ± 0.013	0.4838 ± 0.015	**0.0394 ± 0.001**	**0.2693 ± 0.001**	**0.0306 ± 0.001**
UT-Zappos Temperature=1.3	0.4931 ± 0.012	0.4921 ± 0.013	0.4838 ± 0.015	0.0667 ± 0.003	0.2363 ± 0.002	0.0411 ± 0.001

Table 4.25: The Impact of Temperature scaling on MIT-State: Attribute Accuracy and UCE

Model Configuration	ep1 attr_acc(↑)	ep2 attr_acc(↑)	ep3 attr_acc(↑)	ep1 uce(↓)	ep2 uce(↓)	ep3 uce(↓)
MIT-State	0.2549 ± 0.001	0.2676 ± 0.003	0.2681 ± 0.002	0.1160 ± 0.001	0.3759 ± 0.001	0.4235 ± 0.001
MIT-State Temperature=1.2	0.2549 ± 0.001	0.2676 ± 0.003	0.2681 ± 0.002	**0.022 ± 0.0008**	**0.3514 ± 0.001**	**0.3852 ± 0.0009**
MIT-State Temperature=1.3	0.2549 ± 0.001	0.2676 ± 0.003	0.2681 ± 0.002	0.0205 ± 0.001	0.3448 ± 0.0009	0.3689 ± 0.001
MIT-State Temperature=1.4	0.2549 ± 0.001	0.2676 ± 0.003	0.2681 ± 0.002	**0.0511 ± 0.001**	**0.3415 ± 0.001**	**0.3543 ± 0.0007**

Based on the results from Table 4.26 and Table 4.27, it is found that on UT-Zappos, similar to the performance results with Monte Carlo Dropout, SPE outperforms POE and SOE in terms of performance. Moreover, without model calibration, SPE's performance surpasses other methods.

In line with the above conclusions, according to the results from Table 4.28 and Table 4.29, SPE outperforms POE and SOE in terms of performance. Furthermore, without model calibration, SPE's performance surpasses other methods.

After analyzing these experimental results, **Do we need calibration?** On these two datasets,

we do not need model calibration. Whether adopting Temperature scaling or Monte Carlo dropout for model calibration, the model performance decline.

Table 4.26: Use Temperature scaling for model calibration on UT-Zappos with Accuracy

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 acc(↑)	**0.5357 ± 0.004**	0.5328 ± 0.003	0.5321 ± 0.001	0.5144 ± 0.017	0.5075 ± 0.018	0.5044 ± 0.020
Ep13 acc(↑)	**0.5412 ± 0.005**	0.5321 ± 0.001	0.5306 ± 0.000	0.5150 ± 0.018	0.5085 ± 0.022	0.5041 ± 0.018
Ep23 acc(↑)	**0.5140 ± 0.001**	0.5138 ± 0.000	0.5139 ± 0.001	0.4931 ± 0.019	0.4914 ± 0.016	0.4931 ± 0.019
Ep123 acc(↑)	**0.5372 ± 0.004**	0.5290 ± 0.003	0.5275 ± 0.001	0.5144 ± 0.020	0.5037 ± 0.015	0.5037 ± 0.017

Table 4.27: Use Temperature scaling for model calibration on UT-Zappos with F1 score

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 F1 score(↑)	**0.2922 ± 0.003**	0.2837 ± 0.001	0.2844 ± 0.006	0.2921 ± 0.010	0.2887 ± 0.011	0.2874 ± 0.013
Ep13 F1 score(↑)	**0.2884 ± 0.017**	0.2876 ± 0.008	0.2806 ± 0.007	0.2878 ± 0.011	0.2870 ± 0.015	0.2802 ± 0.012
Ep23 F1 score(↑)	**0.2941 ± 0.005**	0.2844 ± 0.007	0.2839 ± 0.006	0.2788 ± 0.016	0.2788 ± 0.009	0.2798 ± 0.012
Ep123 F1 score(↑)	**0.2977 ± 0.004**	0.2918 ± 0.008	0.2866 ± 0.009	0.2939 ± 0.011	0.2920 ± 0.010	0.2877 ± 0.015

Table 4.28: Use Temperature scaling for model calibration on MIT-States with Accuracy

	Without Model calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 acc(↑)	**0.3056 ± 0.012**	0.2703 ± 0.048	0.2594 ± 0.059	0.3043 ± 0.009	0.3049 ± 0.030	0.3023 ± 0.015
Ep13 acc(↑)	**0.2812 ± 0.018**	0.2660 ± 0.019	0.2543 ± 0.031	0.2798 ± 0.015	0.2785 ± 0.023	0.2672 ± 0.012
Ep23 acc(↑)	**0.3113 ± 0.007**	0.2931 ± 0.024	0.2650 ± 0.005	0.3110 ± 0.013	0.3071 ± 0.015	0.2690 ± 0.016
Ep123 acc(↑)	**0.3110 ± 0.008**	0.2908 ± 0.028	0.2568 ± 0.013	0.3032 ± 0.009	0.3041 ± 0.020	0.2667 ± 0.017

Table 4.29: Use Temperature scaling for model calibration on MIT-States with F1 Score

	Without Model Calibration			With Model Calibration		
	SPE	SOE	POE	SPE	SOE	POE
Ep12 F1 score(↑)	**0.2326 ± 0.003**	0.2168 ± 0.023	0.1962 ± 0.045	0.2317 ± 0.003	0.2310 ± 0.011	0.2227 ± 0.019
Ep13 F1 score(↑)	**0.2286 ± 0.015**	0.2259 ± 0.016	0.2007 ± 0.036	0.2211 ± 0.010	0.2276 ± 0.009	0.2191 ± 0.012
Ep23 F1 score(↑)	**0.2394 ± 0.001**	0.2236 ± 0.005	0.2050 ± 0.015	0.2351 ± 0.011	0.2338 ± 0.013	0.2141 ± 0.004
Ep123 F1 score(↑)	**0.2412 ± 0.003**	0.2349 ± 0.016	0.2116 ± 0.011	0.2410 ± 0.013	0.2390 ± 0.012	0.2198 ± 0.012

Chapter 5

Conclusions and Future Works

In this study, we have successfully developed an innovative ensemble learning approach —Selected Prime Expert (SPE), specifically aimed at optimizing selections of experts under scenarios requiring high confidence and accuracy. SPE, by synthesizing confidence predictions from different experts, not only enhances the predictive performance of a single model but also effectively eliminates outliers, retaining results with smaller prediction errors. This allows for stable performance in situations with high predictive uncertainty or significant fluctuations in errors. Through experimental validation across multiple datasets, our method has demonstrated significant performance improvements under various performance metrics and performed well on the UT-Zappos and MIT-States datasets.

Furthermore, we explored the possibility of replacing the SPE algorithm and assessed the impact of different algorithms on SPE's performance. Based on this, we introduced another ensemble learning strategy—Hierarchical Mixture of Experts (HMOE), which effectively integrates SPE, SOE, and POE ensemble methods. Notably, in some datasets, the HMOE method even surpassed SPE in efficiency and showed advantages over traditional ensemble learning methods in other aspects. Specifically, our method achieved an approximate 2% performance increase on the UT-Zappos dataset and a 3% increase on the MIT-States dataset.

Lastly, this research also delved into whether adopting the SPE method eliminates the need for model calibration and the potential impact of model calibration on performance. After detailed analysis, the results indicate that the model calibration step can be omitted when employing the SPE method. This finding suggests that the SPE method, while enhancing the accuracy of predictions and confidence estimations, inherently adjusts for predictive biases, thereby reducing reliance on traditional model calibration techniques.

For future research, we plan to extend the SPE method to more datasets to evaluate its ap-

plicability and scalability in different scenarios. We also aim to delve deeper into the workings of SPE to more accurately capture its advantages and limitations. Given the diversity of ensemble learning, we will consider integrating other advanced ensemble learning techniques or calibration methods with SPE, hoping to further improve its efficiency and robustness.

References

[1] P. Isola, J. J. Lim, and E. H. Adelson, "Discovering states and transformations in image collections," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1383–1391.

[2] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 192–199.

[3] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.

[4] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[6] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[7] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[8] R. Polikar, "Ensemble learning," *Ensemble machine learning: Methods and applications*, pp. 1–34, 2012.

[9] E. S. Aimar, A. Jonnarth, M. Felsberg, and M. Kuhlmann, "Balanced product of experts for long-tailed recognition," *arXiv preprint arXiv:2206.05260*, 2022.

[10] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, no. 1-3, pp. 65–96, 1984.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 951–958.

[12] I. Misra, A. Gupta, and M. Hebert, "From red wine to red tomato: Composition with context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1792–1801.

[13] N. Saini, K. Pham, and A. Shrivastava, "Disentangling visual embeddings for attributes and objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 658–13 667.

[14] S. Kumar, A. Iftekhar, E. Prashnani, and B. Manjunath, "Locl: Learning object-attribute composition using localization," *arXiv preprint arXiv:2210.03780*, 2022.

[15] X. Li, X. Yang, K. Wei, C. Deng, and M. Yang, "Siamese contrastive embedding network for compositional zero-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 9326–9335.

[16] X. Lu, S. Guo, Z. Liu, and J. Guo, "Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 560–23 569.

[17] C. Wang, C. Lawrence, and M. Niepert, "Uncertainty estimation and calibration with finite-state probabilistic rnns," *arXiv preprint arXiv:2011.12010*, 2020.

[18] M. Huang and Y. Qiao, "Uncertainty-estimation with normalized logits for out-of-distribution detection," in *International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2023)*, vol. 12645. SPIE, 2023, pp. 524–530.

[19] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730.

[20] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[21] M.-H. Laves, S. Ihler, K.-P. Kortmann, and T. Ortmaier, "Uncertainty calibration error: A new metric for multi-class classification," 2020.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.