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

For oral defence

Reporter：周峻緯

Date：2024/03/27

CONTENTS

Introduction

PART ONE

Part1 : Introduction
- Task description

Goal

- Performing attribute-only prediction in Compositional Zero-Shot Learning.

Challenge

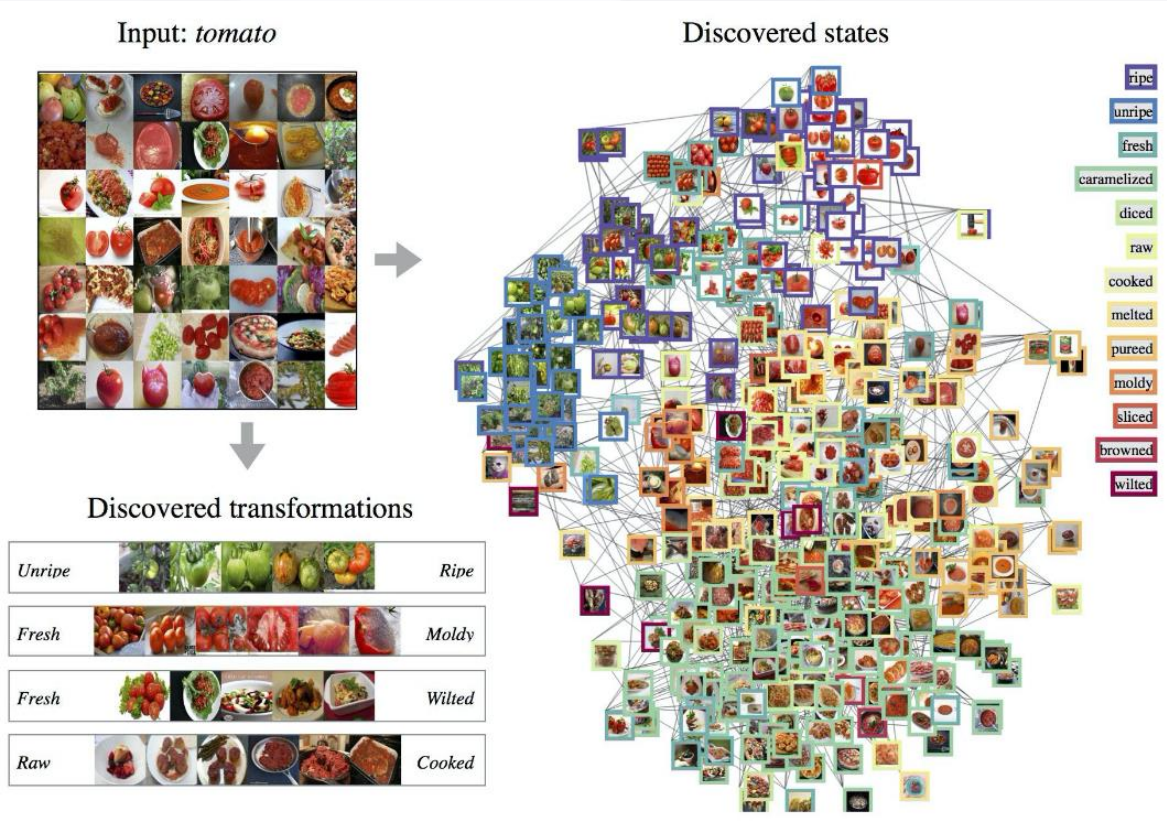
- There are many methods (aka experts) to do that. But how to effectively integrate multiple experts?

Solution

- Define a new expert, dubbed Selected Prime Expert(SPE), based uncertainty-to-error-rate mapping, then integrate it with sum of experts(SOE) and products of experts(POE) by a Hierarchical mixture of experts (HMOE) technology.

Part1 : Introduction

- Dataset(Compositional zero-shot learning)



Dataset	MIT-States
Image Count	~50,000
Attribute Count	245
Object Count	115
Attribute Examples	mossy, deflated, dirty
Object Examples	fish, persimmon, room

Part1 : Introduction

- Dataset(Compositional zero-shot learning)

Dataset	UT-Zappos
Image Count	~29000
Attribute Count	115
Object Count	14
Attribute Examples	pointy, sporty, comfort
Object Examples	shoes, sandals, slippers, boots

Part1 : Introduction
- Dataset split

Y^S :The number of seen pairs Y^u :The number of unseen pairs

n_{att}: The number of attributes. n_{obj}: The number of objects.

	Train set			Val set		Test set	
Datasets	n_{att}	n_{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

In open-world settings :

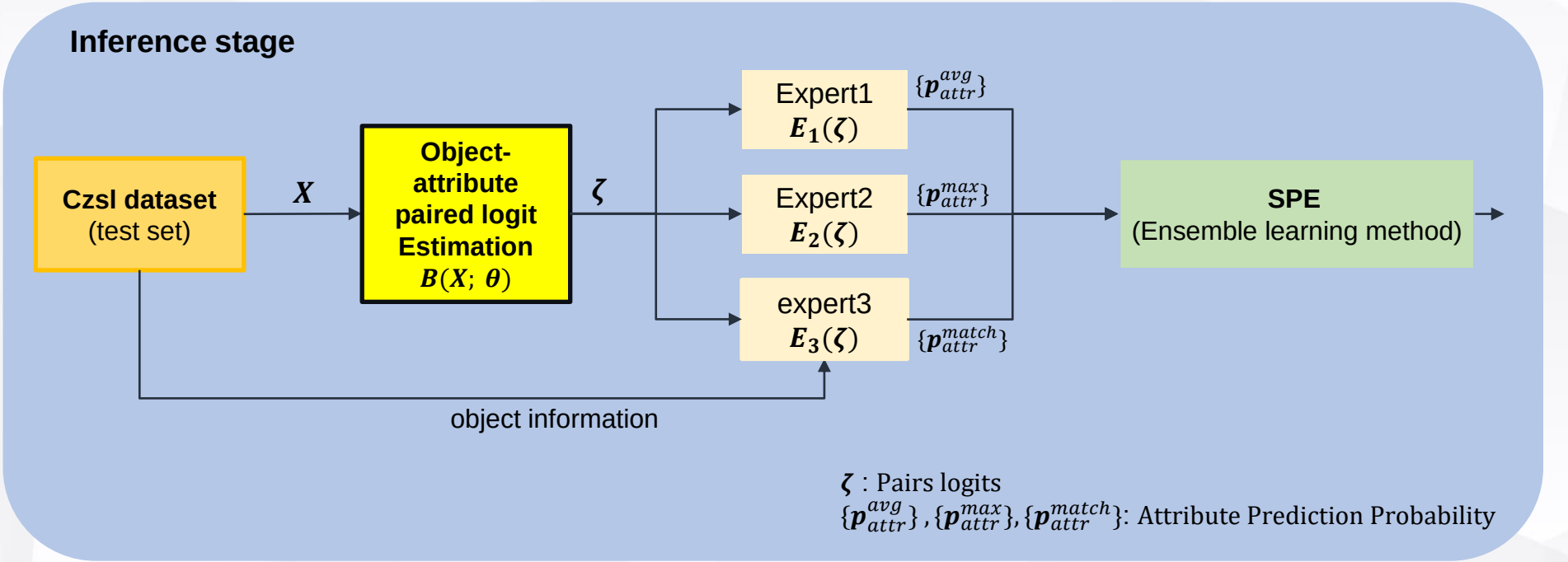
$n_{pair} = n_{att} * n_{obj}$

Train set, validation set, test set are mutual exclusive.

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

Part1 : Introduction

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



Our method improves performance by approximately 2% on the UT-Zappos dataset and by approximately 3% on the MIT-States dataset in attribute-only prediction.

Part1: Introduction
- Summary

- SPE, our proposed method, effectively resolves model uncertainty issues. It does this without depending on traditional model calibration techniques like dropout or temperature scaling.

- The application of HMOE allows the integration of the capabilities of SOE, POE and SPE. This integration results in improved overall performance.

Dataset: UT-Zappos	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE
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
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

Dataset: MIT-States	SPE	HMOE-SOE	HMOE-POE	HMOE-SPE	SOE	POE
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
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

Related works

PART TWO

Part2: Related works
-Ensemble learning

[1]. G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural computation, vol. 14, no. 8, pp. 1771–1800, 2002.
[2]. Polikar, “Ensemble learning,” Ensemble machine learning: Methods and applications,pp. 1–34, 2012.

- Ensemble learning is a technique that enhances prediction and generalization capabilities by combining multiple machine learning models.
- The main concept of this strategy is that although a single model may overfit the training data, combining multiple models can reduce such errors.
- By integrating models, ensemble learning can improve prediction accuracy and reduce the risk of overfitting.
- Common ensemble learning methods include **sum of experts, product of experts[1], simple voting, and weighted voting[2]**.

	SPE	Other Ensemble learning
Goal	By integrating predictions from multiple models, the aim is to enhance the accuracy and reliability of the predictions and optimize the overall performance of the model.	
Method	SPE chooses the optimal model prediction through Normalized Uncertainty and UCE(Uncertainty Calibration Error).	These methods mainly integrate outputs from multiple models, via methods such as weighted averaging or multiplication.
Advantages	SPE selects known high-performance models, avoiding the influence of inferior models, to achieve optimal predictions.	Even when the best model is indeterminate, other ensemble learning methods enhance model stability and generalizability through combined predictions.

Part2: Related works
-Compositional Zero-Shot Learning(CZSL)

[3] D. D. Hoffman and W. A. Richards, “Parts of recognition,” Cognition, vol. 18, no. 1-3, pp.65–96, 1984
[4] 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.

- Compositional Zero-Shot Learning (CZSL) was first introduced in the paper "Recognition by Parts" [3]. This method enables the identification of objects that have not been previously encountered.
- In the traditional approach to Zero-Shot Learning (ZSL), the model is capable of recognizing images from known categories, as well as descriptions of both known and unknown categories.
- Unlike ZSL, CZSL does not provide descriptions of any known or unknown attributes and object compositions. However, all attribute and object concepts are known during the training phase.

	Our task	(Decomposed Fusion with Soft Prompt)[4] DFSP
Goal	Our methods involves predicting attributes when the object is already known.	DFSP predicts the pairings between known objects and attributes.

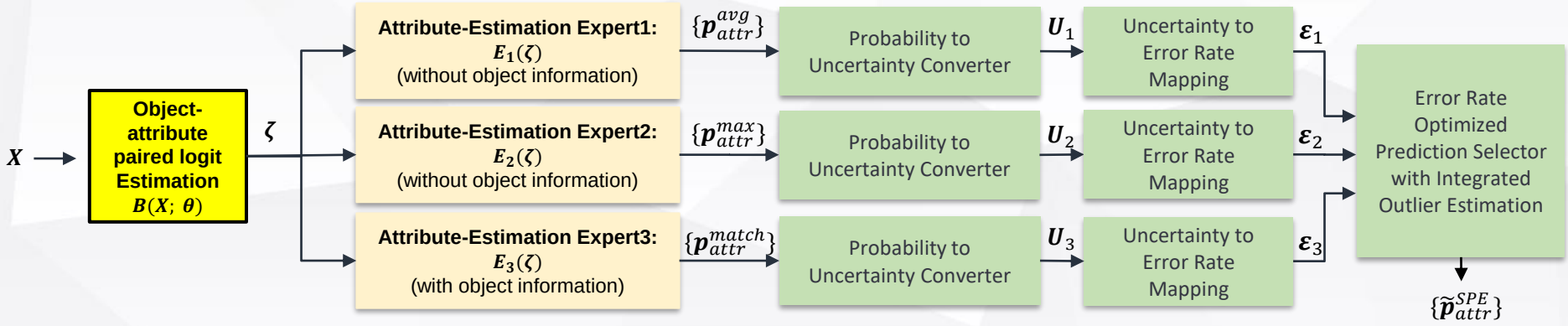
Method

PART THREE

Part3 : Method

- Selected Prime Expert(SPE) Inference Stage Architecture Diagram

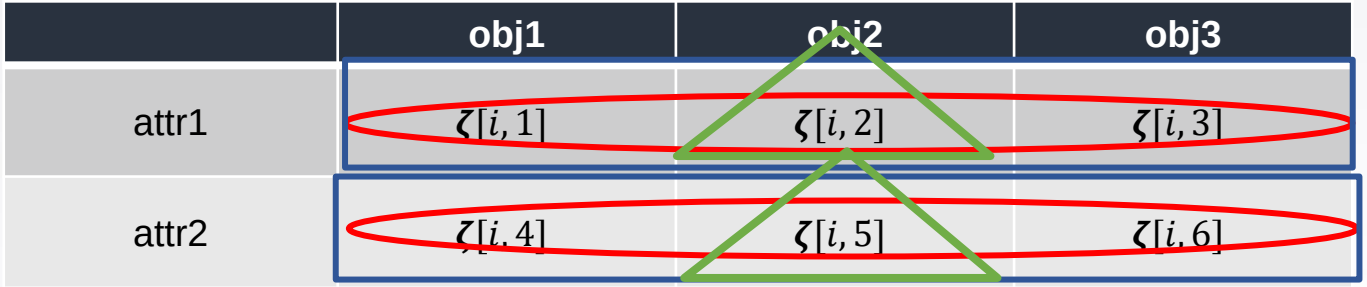
1. Input images $\boldsymbol{X}$: In the test dataset, there are n images, and our set $\boldsymbol{X}$ is composed of these n images.
2. Object-attribute paired logit Estimation $\boldsymbol{B}(\boldsymbol{X};\ \boldsymbol{\theta})$: The Decomposed Fusion with Soft Prompt (DFSP) model is base model that generate input of SPE and HMOE(Output: $\boldsymbol{\zeta}$)
3. Model output logit $\boldsymbol{\zeta}$: The logits are the output of $\boldsymbol{B}(\boldsymbol{X};\ \boldsymbol{\theta})$.
4. Expert 1 $\boldsymbol{E_1}(\boldsymbol{\zeta})$: Integrate all identical attributes and calculate the average of the logit for each attribute. (Output:$\boldsymbol{p}_{attr}^{avg}$)
5. Expert 2 $\boldsymbol{E_2}(\boldsymbol{\zeta})$: For each sample, the highest value is chosen from each attribute category. (Output: $\boldsymbol{p}_{attr}^{max}\}$)
6. Expert 3 $\boldsymbol{E_3}(\boldsymbol{\zeta})$: Once the information of the object is known, find out the corresponding attributes. (Output: $\boldsymbol{p}_{attr}^{match}$)



Part3 : Method

- Attribute-Estimation Expert overview

- In the test dataset, there are n images, and our set $\boldsymbol{X}$ is composed of these n images.
- We use the symbol $\boldsymbol{\zeta}[i,j]$ to represent the j-th element associated with the i-th image, where there are n such images in total.
- The goal is to convert the Logit $\boldsymbol{\zeta}$ used for n_{pair} forecasting to n_{attr}, which means that it will now predict all attribute categories instead of just pairs.
- Three experts have each adopted different methods to convert $\boldsymbol{\zeta}$.



Attribute-Estimation Expert 1 $\boldsymbol{E_1(\zeta)}$ ○ (average) : $\boldsymbol{\zeta}_{attr}^{avg}[i, attr1] = (\boldsymbol{\zeta}[i,1]+\boldsymbol{\zeta}[i,2]+\boldsymbol{\zeta}[i,3])/3$

$\boldsymbol{\zeta}_{attr}^{avg}[i, attr2] = (\boldsymbol{\zeta}[i,4]+\boldsymbol{\zeta}[i,5]+\boldsymbol{\zeta}[i,6])/3$

Attribute-Estimation Expert 2 $\boldsymbol{E_2(\zeta)}$ □ (argmax) : $\boldsymbol{\zeta}_{attr}^{max}[i, attr1] = argmax_i(\boldsymbol{\zeta}[i,1], \boldsymbol{\zeta}[i,2], \boldsymbol{\zeta}[i,3])$

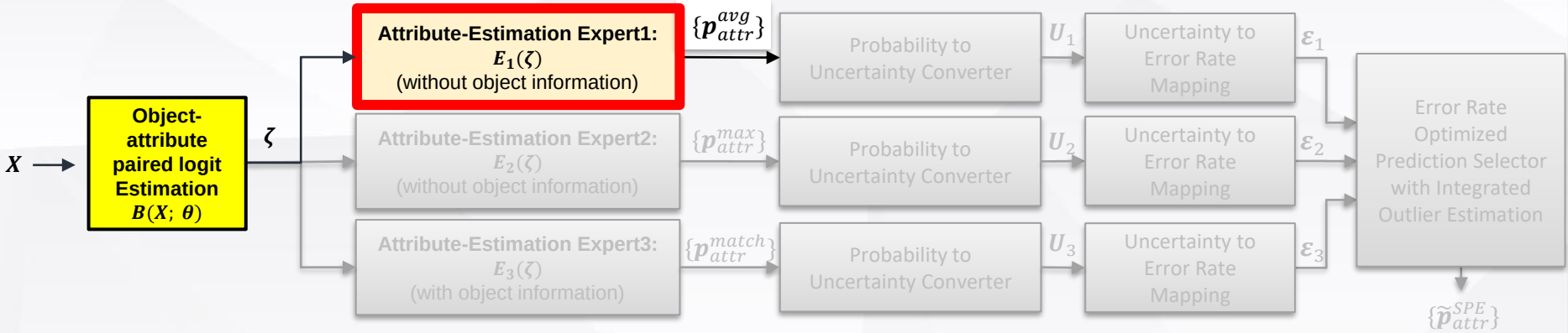
$\boldsymbol{\zeta}_{attr}^{max}[i, attr2] = argmax_i(\boldsymbol{\zeta}[i,4], \boldsymbol{\zeta}[i,5], \boldsymbol{\zeta}[i,6])$

Attribute-Estimation Expert 3 $\boldsymbol{E_3(\zeta)}$ △ (match) : $\boldsymbol{\zeta}_{attr}^{match}[i\ , attr1] = \boldsymbol{\zeta}[i,2]$ *when obj equal obj2*

$\boldsymbol{\zeta}_{attr}^{match}[i\ , attr2] = \boldsymbol{\zeta}[i,5]$ *when obj equal obj2*

$\boldsymbol{p}_{attr}^{avg} = \boldsymbol{softmax}(\boldsymbol{\zeta}_{attr}^{avg})$, $\boldsymbol{p}_{attr}^{max} = \boldsymbol{softmax}(\boldsymbol{\zeta}_{attr}^{max})$, $\boldsymbol{p}_{attr}^{match} = \boldsymbol{softmax}(\boldsymbol{\zeta}_{attr}^{match})$

Part3 : Method
- Attribute-Estimation Expert1

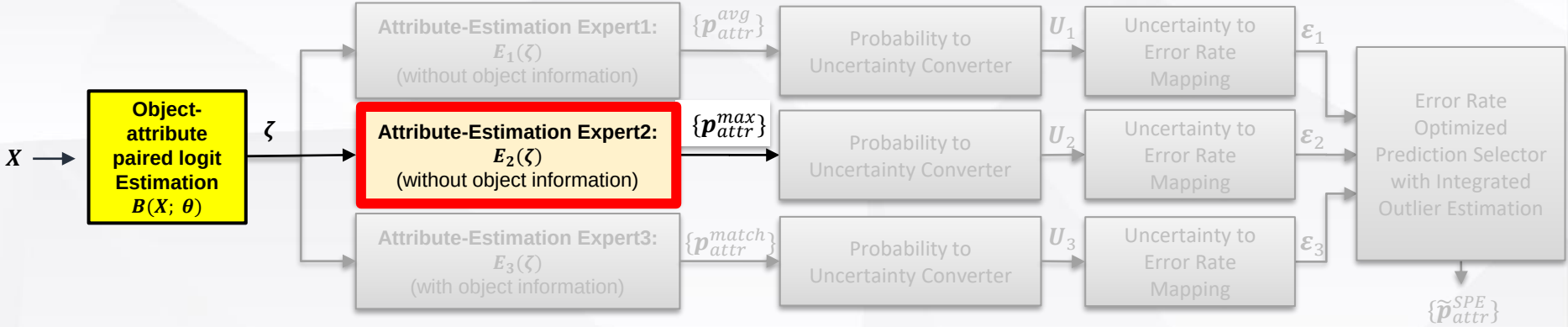


- Expert1's approach is to aggregate the logit pairs that belong to a specific attribute and calculate the average by dividing the sum by the number of objects.

	obj1	obj2	obj3
attr1	$\zeta[i,1]$	$\zeta[i,2]$	$\zeta[i,3]$
attr2	$\zeta[i,4]$	$\zeta[i,5]$	$\zeta[i,6]$

$$\zeta_{attr}^{avg}[i,attr1] = (\zeta[i,1]+\zeta[i,2]+\zeta[i,3])/3$$
$$\zeta_{attr}^{avg}[i,attr2] = (\zeta[i,4]+\zeta[i,5]+\zeta[i,6])/3$$
$$p_{attr}^{avg} = softmax(\zeta_{attr}^{avg})$$

Part3 : Method
- Attribute-Estimation Expert2



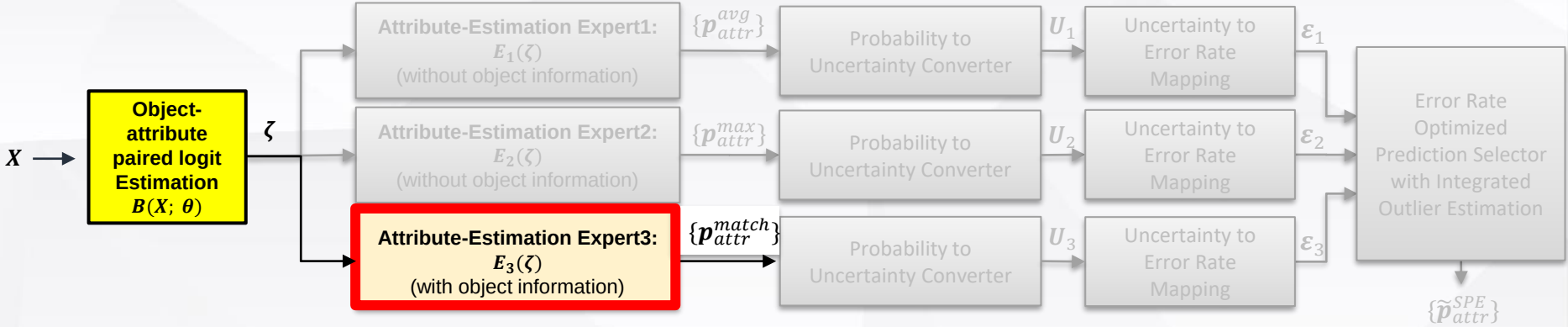
- Expert 2 uses a method that selects the maximum values in each attribute category from the model output logit matrix $\boldsymbol{\zeta}$ for each sample.

	obj1	obj2	obj3
attr1	$\zeta[i,1]$	$\zeta[i,2]$	$\zeta[i,3]$
attr2	$\zeta[i,4]$	$\zeta[i,5]$	$\zeta[i,6]$

$$\zeta_{attr}^{max}[i,attr1] = argmax_i(\zeta[i,1],\zeta[i,2],\zeta[i,3])$$
$$\zeta_{attr}^{max}[i,attr2] = argmax_i(\zeta[i,4],\zeta[i,5],\zeta[i,6])$$

$$\boldsymbol{p_{attr}^{max} = softmax(\zeta_{attr}^{max})}$$

Part3 : Method
- Attribute-Estimation Expert3



- Assuming the true class of the i-th image is **(attr1, obj2).**

	obj1	obj2	obj3
attr1	$\zeta[i,1]$	$\zeta[i,2]$	$\zeta[i,3]$
attr2	$\zeta[i,4]$	$\zeta[i,5]$	$\zeta[i,6]$

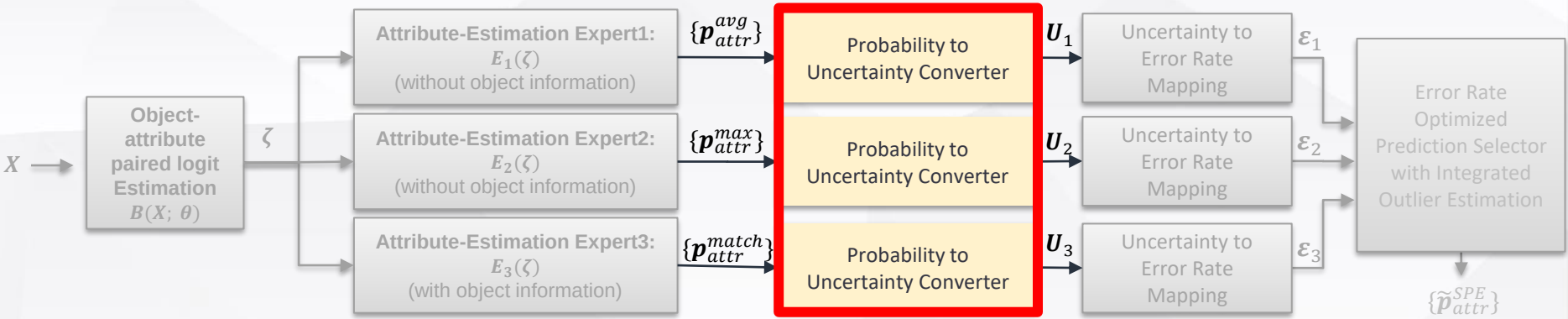
- We calculate ζ_{attr}^{match} based on the information of the object because the true class of this image is obj2. We search for the corresponding logit pair and insert the logit into the corresponding ζ_{attr}^{match}.

$$\zeta_{attr}^{match}[i, attr1] = \zeta[i,2] \quad when\ obj\ equal\ obj2$$
$$\zeta_{attr}^{match}[i, attr2] = \zeta[i,5] \quad when\ obj\ equal\ obj2$$

$$p_{attr}^{match} = \boldsymbol{softmax}(\zeta_{attr}^{match})$$

Part3 : Method

- SPE: Probability to Uncertainty Converter



- We calculated the information entropy of the probability distributions for three experts as a measure of uncertainty. The larger the entropy, the higher the uncertainty.
- The formula for calculating entropy is $H(\boldsymbol{p}_i) \coloneqq \frac{1}{log\ C}\sum_{c=1}^{C} -\boldsymbol{p}^{(c)} log \boldsymbol{p}^{(c)}$, $H(\boldsymbol{p}^1)\in[0,1]$. , where p_i is each probability value in the distribution. I define all attribute categories by N_a.
- In our example, the probability distributions for three experts are:

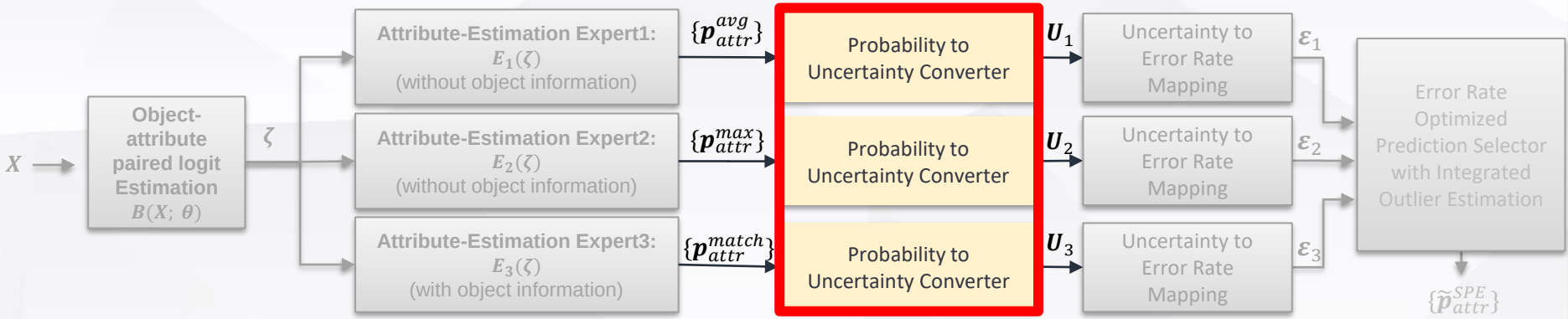
$$\boldsymbol{U_1} = \frac{1}{log\ N_a}\sum_{i=1}^{N_a} -(\boldsymbol{p}_{attr}^{avg})_i \cdot log(\boldsymbol{p}_{attr}^{avg})_i \quad ,\mathrm{H}(\boldsymbol{p}_{attr}^{avg}) \in[0,1].$$

$$\boldsymbol{U_2} = \frac{1}{log\ N_a}\sum_{i=1}^{N_a} -(\boldsymbol{p}_{attr}^{max})_i \cdot log(\boldsymbol{p}_{attr}^{max})_i \quad ,\mathrm{H}(\boldsymbol{p}_{attr}^{max})\in[0,1].$$

$$\boldsymbol{U_3} = \frac{1}{log\ N_a}\sum_{i=1}^{N_a} -(\boldsymbol{p}_{attr}^{match})_i \cdot log(\boldsymbol{p}_{attr}^{match})_i \quad ,\mathrm{H}(\boldsymbol{p}_{attr}^{match})\in[0,1].$$

Part3 : Method

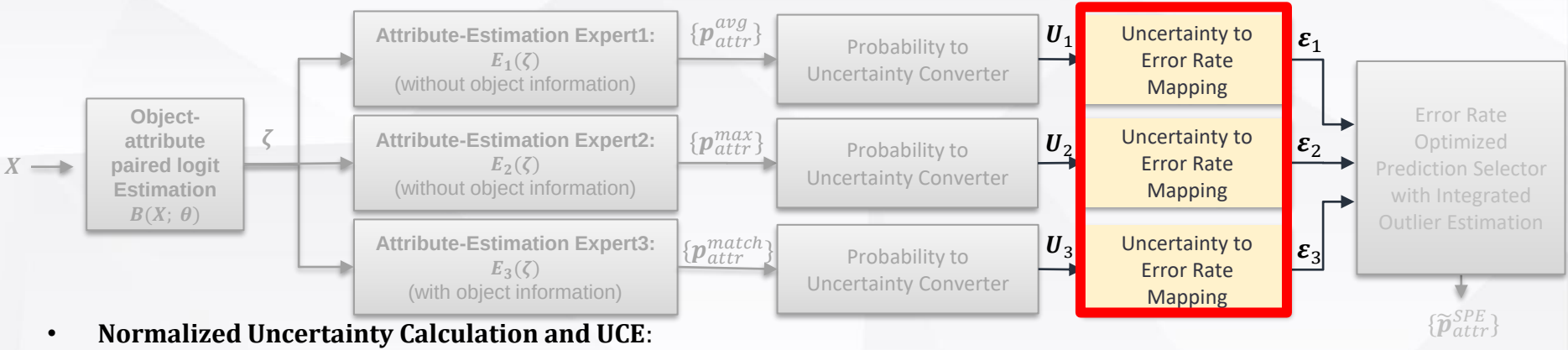
- SPE: Probability to Uncertainty Converter



- In the inference stage, we utilize the normalized uncertainties $\boldsymbol{U_1}$, $\boldsymbol{U_2}$ and, $\boldsymbol{U_3}$ from all experts to find the corresponding mapped error rates on the Uncertainty to Error Rates mapping table established during the training phase (see next page for details).

Part3 : Method

- SPE: Uncertainty to Error Rate Mapping



- **Normalized Uncertainty Calculation and UCE**:

During the training phase, our method calculates the Uncertainty Calibration Error (UCE) by assessing the normalized uncertainty of each model's probability output and establishes the Uncertainty to Error Rates Mapping. Please refer to the next page for further details.

- **Mapping Table Establishment and Utilization**:

We subsequently construct a Mapping table grounded on Uncertainty and Error Rates. This table is applied in the testing phase for respective assessments.

$$U_1 \rightarrow \varepsilon_1, \quad \varepsilon_1 \in [0,1]$$
$$U_2 \rightarrow \varepsilon_2, \quad \varepsilon_2 \in [0,1]$$
$$U_3 \rightarrow \varepsilon_3, \quad \varepsilon_3 \in [0,1]$$

Part3 : Method

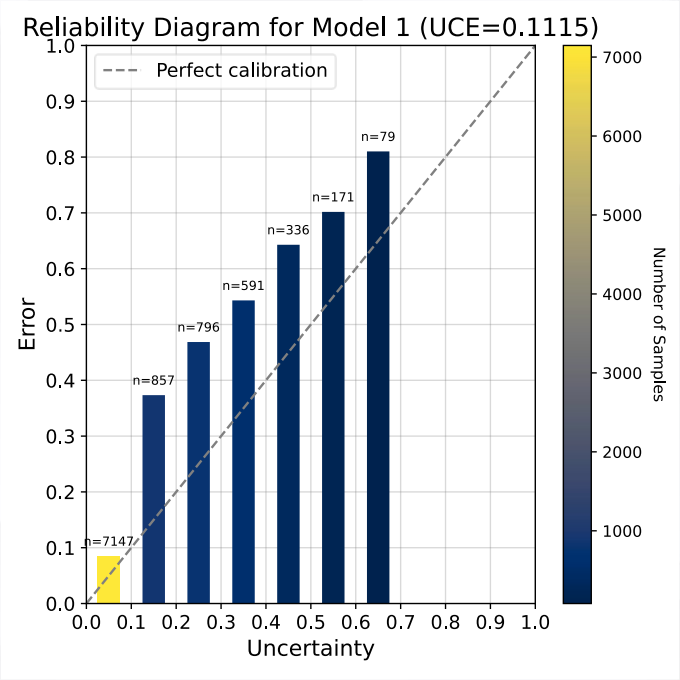
- SPE: Uncertainty to Error Rates Mapping reliability diagram

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

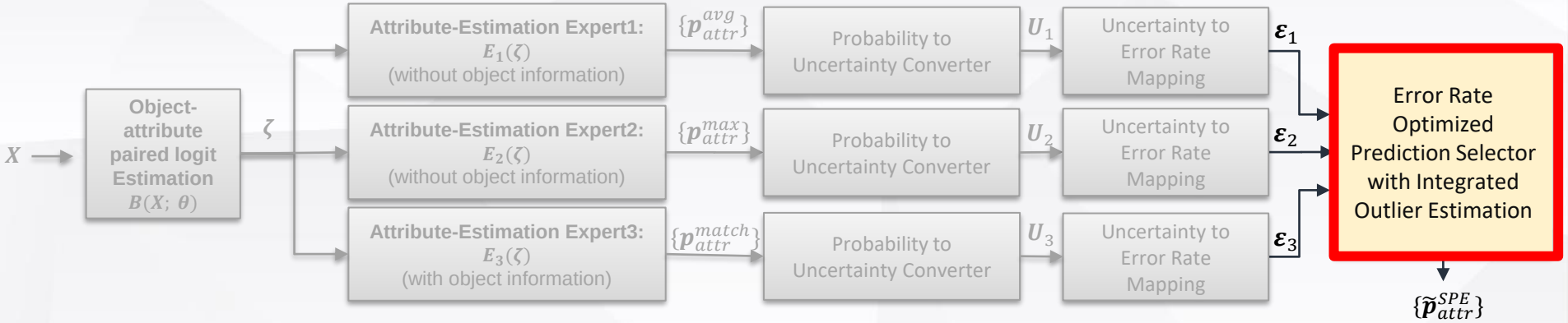
- The U to E Mapping reliability diagram will be used during the inference stage.

- If the model is entirely reliable, the diagram would form a 45-degree line, denoting perfect alignment between uncertainty and error rate.

- This diagram uses color to represent the quantity of samples. Yellow indicates a larger number of samples, while blue indicates a smaller number.



Part3 : Method
- SPE Outlier estimation

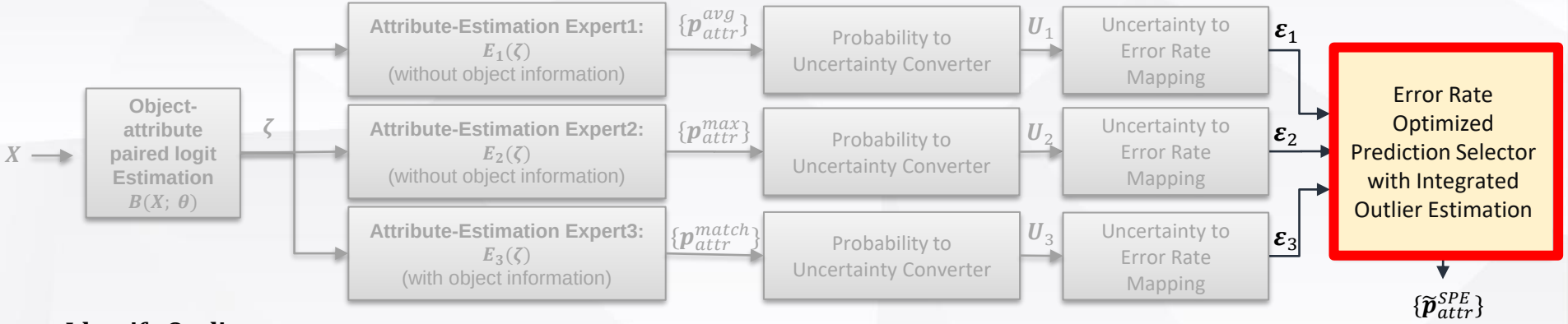


Error Rate Optimized Prediction Selector with Integrated Outlier Estimation

- After inputting three error rates into the Error Rate Optimized Prediction Selector with Integrated Outlier Estimation, outliers are excluded. Based on the remaining number of error rates , the system selects either the best probability or the average probability. Finally, it outputs $\{\tilde{p}_{attr}^{SPE}\}$.

Part3 : Method

- SPE : Error Rate Optimized Prediction Selector with Integrated Outlier Estimation



- **Identify Outliers**

Calculate the absolute difference between the error rate of each sample and the minimum error rate. If this difference is less than the defined threshold θ, then the sample is identified as an outlier. Here, θ represents the threshold used for identifying outliers.

For the i-th sample, calculate the **minimum error rate**:

$$\varepsilon_{min_i} = min(\varepsilon_{1_i}, \varepsilon_{2_i}, \varepsilon_{3_i})$$

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

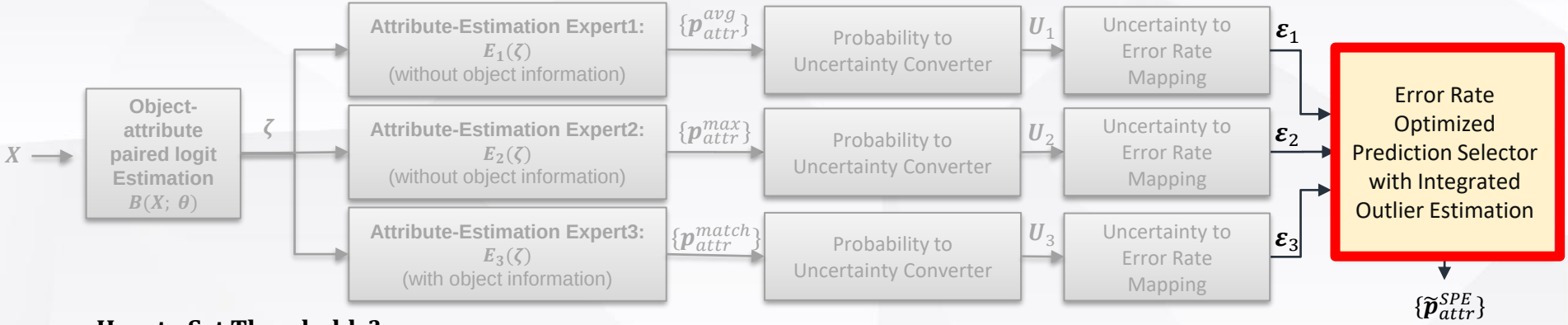
$$\Delta\varepsilon_{j_i} = |\varepsilon_{j_i} - \varepsilon_{min_i}|$$

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

$$\Delta\varepsilon_{j_i} < \theta$$

then determine that the error rate for that sample for that expert is an outlier.

Part3 : Method
- SPE Outlier estimation



- **How to Set Thresholds?**

We will experiment with different threshold combinations on the validation dataset to find the optimal threshold. This optimal threshold will then be applied to the test dataset.

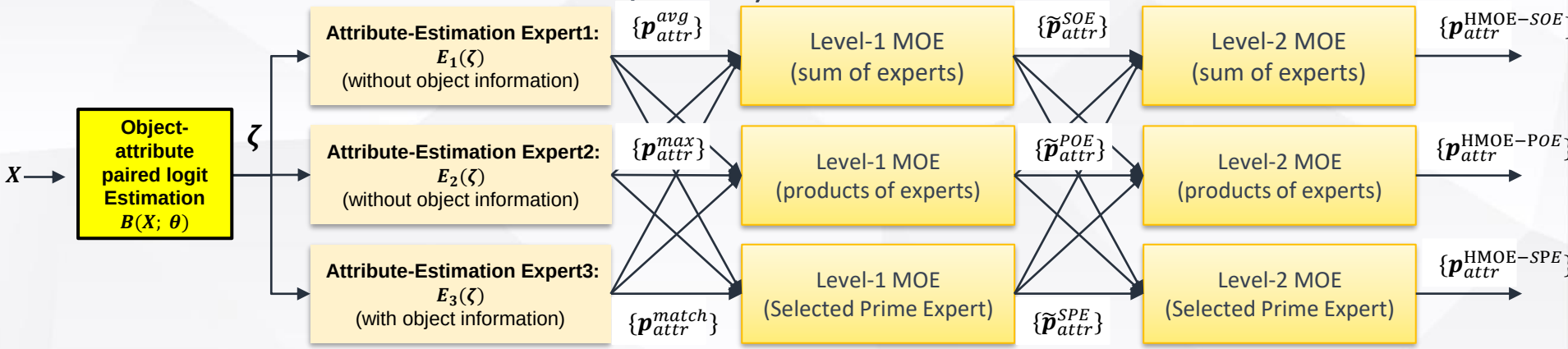
- **Expert Probability Selection Based on Error Rate Thresholding**

P_{j_i} and P_{k_i} and P_{l_i} are the predicted probabilities for the i-th sample from experts j , k and l respectively.

$$\{\tilde{p}_{attr}^{SPE}\} = \begin{cases} \frac{P_{j_i}+P_{k_i}+P_{l_i}}{3} & if\ three\ experts\ j,k,l\ satisfy\ \Delta\varepsilon_{j_i} < \theta\ and\ \Delta\varepsilon_{k_i} < \theta\ and\ \Delta\varepsilon_{l_i} < \theta \\ \frac{P_{j_i}+P_{k_i}}{2} & if\ two\ experts\ j,k\ satisfy\ \Delta\varepsilon_{j_i} < \theta \text{ and } \Delta\varepsilon_{k_i} < \theta \\ P_{j_i} & if\ one\ expert\ j\ satisfies\ \Delta\varepsilon_{j_i} < \theta \end{cases}$$

Part3 : Method

- Our method: Hierarchical MOE(HMOE)



The text describes the probability outputs from three experts, denoted as $\boldsymbol{p}_{attr}^{avg}$, $\boldsymbol{p}_{attr}^{max}$ and $\boldsymbol{p}_{attr}^{match}$. We initially compute $\boldsymbol{p}^{SPE}$ using our SPE method. Subsequently, we calculate $\boldsymbol{p}^{SOE}$ and $\boldsymbol{p}^{POE}$ using the **sum of experts**(SOE) and **products of experts**(POE) methods, respectively.

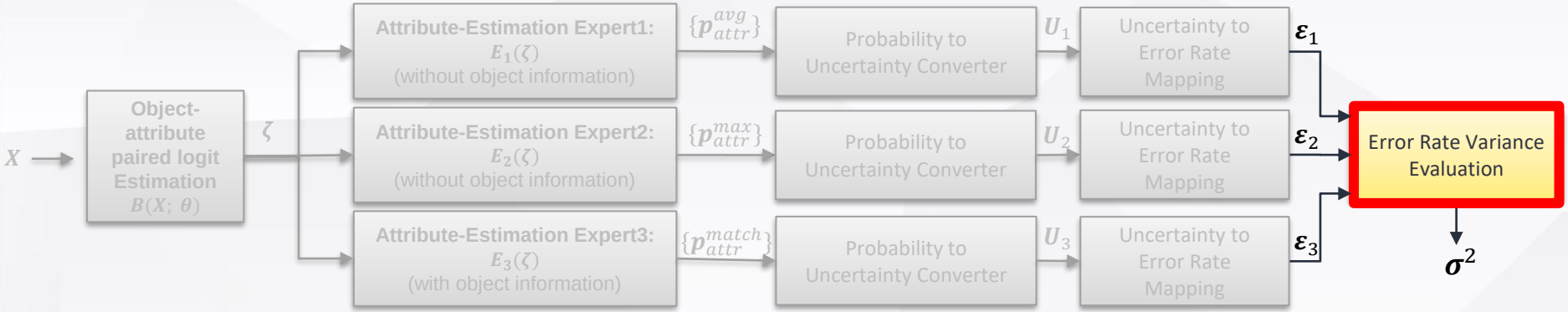
$$\tilde{\boldsymbol{p}}_{attr}^{SPE} = f(\boldsymbol{p}_{attr}^{avg}, \boldsymbol{p}_{attr}^{max}, \boldsymbol{p}_{attr}^{match}),\ \tilde{\boldsymbol{p}}_{attr}^{SOE} = \frac{(\boldsymbol{p}_{attr}^{avg} + \boldsymbol{p}_{attr}^{max} + \boldsymbol{p}_{attr}^{match})}{3},\ \tilde{\boldsymbol{p}}_{attr}^{POE} = (\boldsymbol{p}_{attr}^{avg} * \boldsymbol{p}_{attr}^{max} * \boldsymbol{p}_{attr}^{match}).$$

Finally, we will use the SPE method, along with the SOE and POE methods, to calculate $\boldsymbol{p}_{attr}^{SOP}$, $\boldsymbol{p}_{attr}^{SOE}$, and $\boldsymbol{p}_{attr}^{SOE}$.

$$\boldsymbol{p}_{attr}^{\mathrm{HMOE}-SPE} = f(\tilde{\boldsymbol{p}}_{attr}^{\mathrm{SPE}}, \tilde{\boldsymbol{p}}_{attr}^{\mathrm{POE}}, \tilde{\boldsymbol{p}}_{attr}^{\mathrm{SOE}}),\ \boldsymbol{p}_{attr}^{\mathrm{HMOE}-SOE} = \frac{(\tilde{\boldsymbol{p}}_{attr}^{\mathrm{SPE}} + \tilde{\boldsymbol{p}}_{attr}^{\mathrm{POE}} + \tilde{\boldsymbol{p}}_{attr}^{\mathrm{SOE}})}{3},\ \boldsymbol{p}_{attr}^{\mathrm{HMOE}-POE} = (\tilde{\boldsymbol{p}}_{attr}^{\mathrm{SPE}} * \tilde{\boldsymbol{p}}_{attr}^{\mathrm{POE}} * \tilde{\boldsymbol{p}}_{attr}^{\mathrm{SOE}}).$$

Part3 : Method

- Evaluating the Variance of the Average Error Rate



The variance of the error rates was calculated for each sample by these three experts. The index i represents the number of samples.

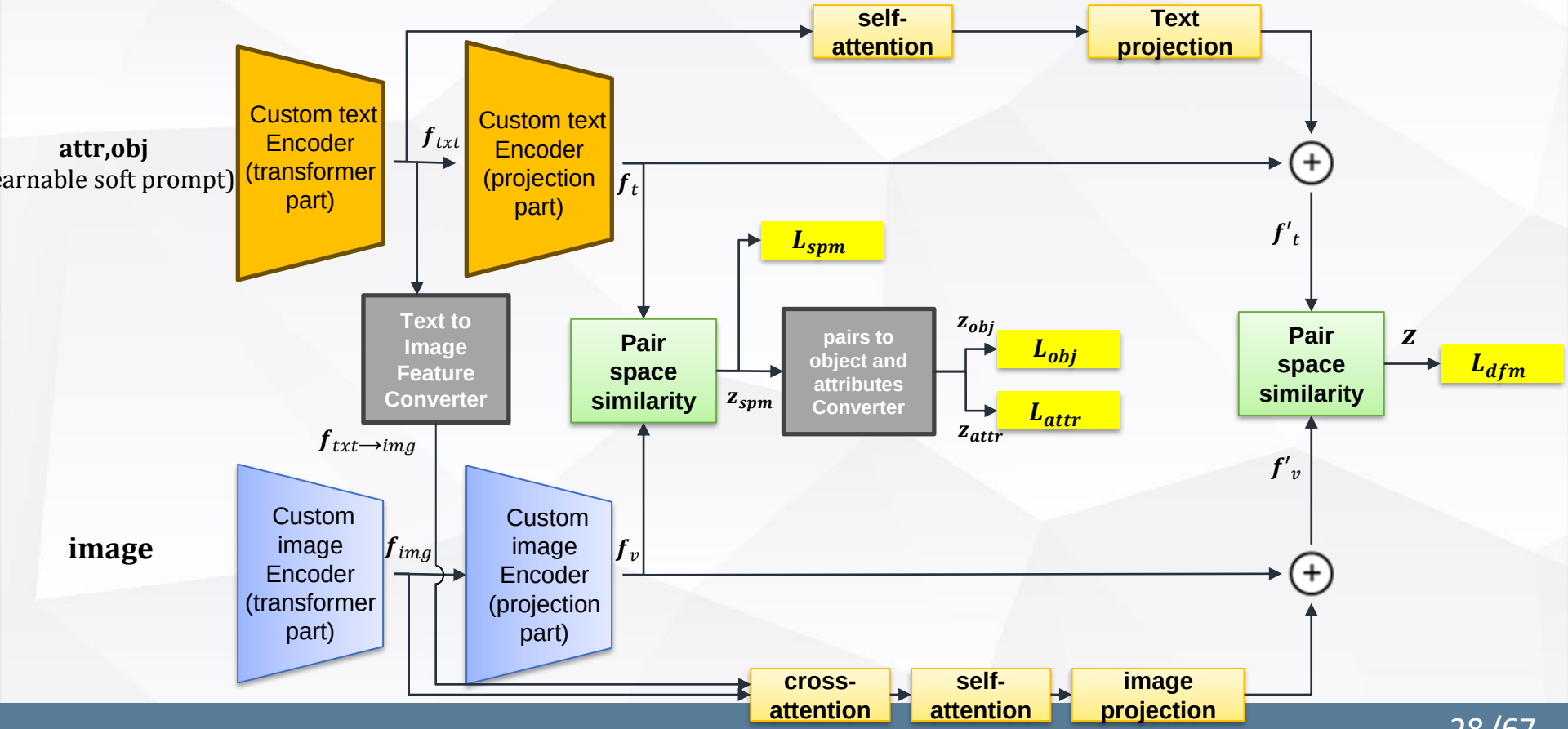
$$\sigma_i^2 = \frac{1}{3}((\varepsilon_{1_i} - \bar{\varepsilon}_i) + (\varepsilon_{2_i} - \bar{\varepsilon}_i) + (\varepsilon_{3_i} - \bar{\varepsilon}_i)) \qquad \bar{\varepsilon}_i = \frac{\varepsilon_{1_i} + \varepsilon_{2_i} + \varepsilon_{3_i}}{3}$$

The variance of the error rate for each sample, denoted as σ_{avg}^2, can be calculated by taking the average of the variances.

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

Part3 : Method

- Logit Estimation Model Architecture(Compositional zero-shot learning)



Experimental results

PART FOUR

Part4 : Experiment
- Experiment Setup

Table 3.1. Summary for the experiment setup.

Framework	PyTorch
Backbone	Clip (pretrained on LAION-400M)
Image size	3 × 224 × 224 (channel × height × width)
Epoch	20
Batch size	128
Learning rate	$lr_{init} = 0.0005$
Optimizer	Adam Optimizer
GPU	A single NVIDIA Tesla V100 GPU for each experiment

Part4 : Experiment
- Evaluation Metrics : Confusion Matrix

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

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

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

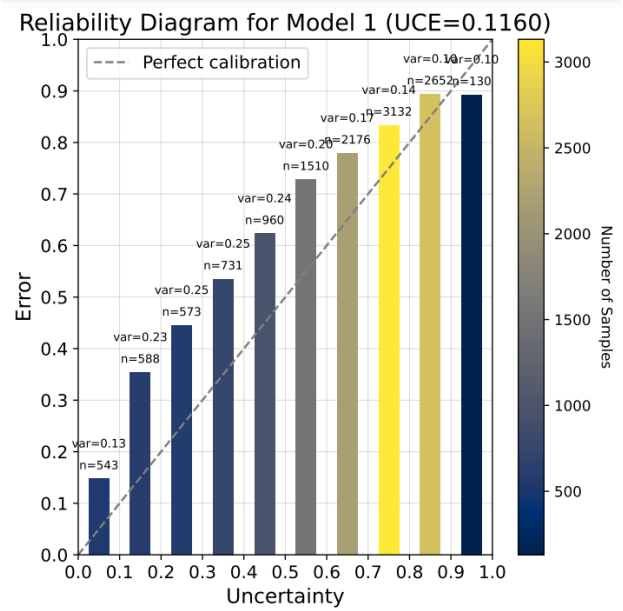
$$\text{Precision} = \frac{TP}{FP + TP}$$

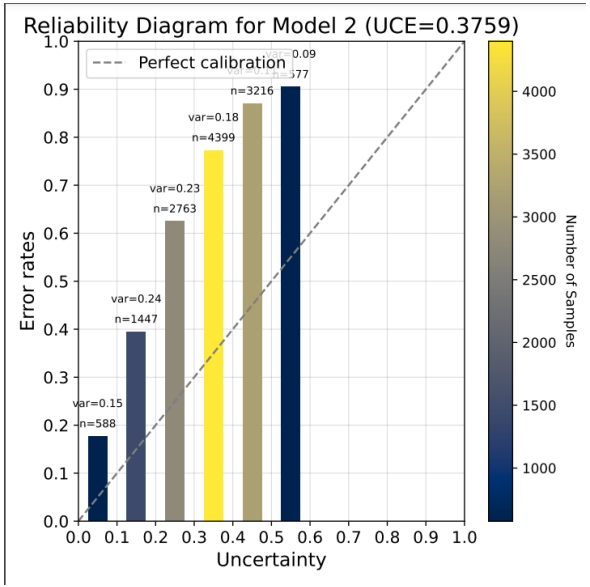
$$\text{Recall} = \frac{TP}{FN + TP}$$

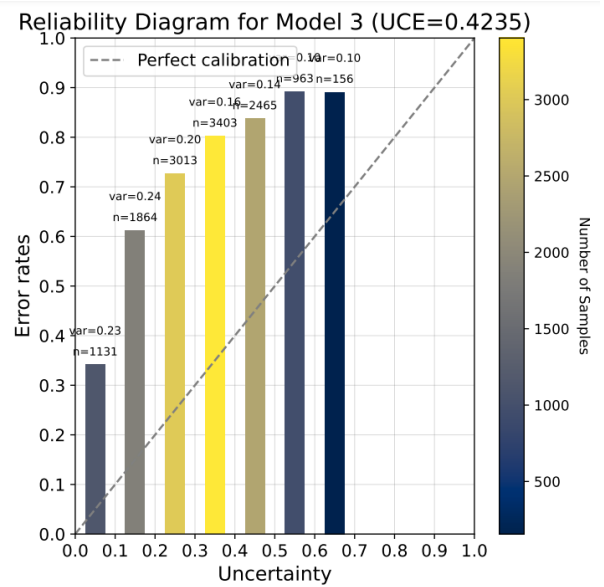
$$\text{F1 Score} = \frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Part4 : Experiment
-SPE Uncertainty to Error Rates Mapping reliability diagram

Dataset: MIT-States

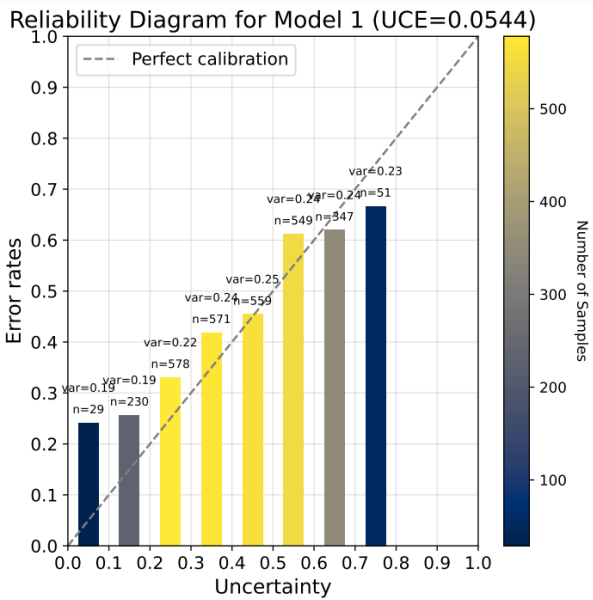


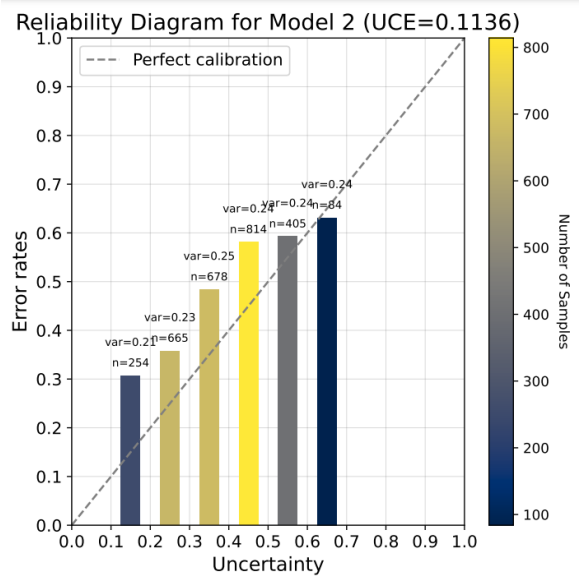


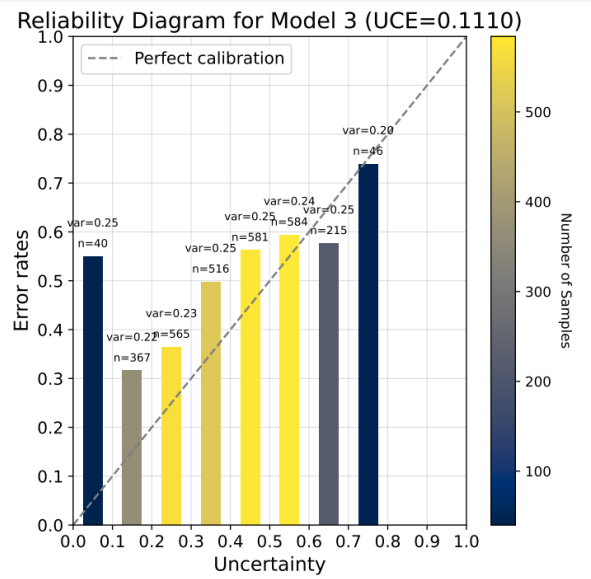


Part4 : Experiment
-SPE Uncertainty to Error Rates Mapping reliability diagram

Dataset: UT-Zappos







- SPE with Dynamic Thresholding

- The primary objective is to select a threshold based on error rates to exclude outliers, for studying the variations in the SPE Outlier Approach under different threshold conditions.

Part4 : Experiment

- SPE with Dynamic Thresholding

Dataset: UT-Zappos

	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

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

Part4 : Experiment

- SPE with Dynamic Thresholding

Dataset: MIT-States

	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

Threshold	SPE ep123 acc(↑)
0.005	0.3104 ±0.008
0.001	0.3106 ±0.008
0.05	**0.3110 ±0.008**
0.01	0.3098 ±0.011
0.1	0.3095 ±0.011

Experimental Results

- Based on these results, it can be observed that on the UT-Zappos and MIT-States dataset, **SPE** outperforms **HMOE-SOE**, achieving the best performance.

Comparison on Other Ensemble Learning Methods

- Our approach involves predicting attributes in the context where the object is already known.
- To make predictions, we will use a Logit Estimation Model and test our method. This comparison will help us identify any differences between our approach and previous conventional ensemble learning methods.

Part4 : Experiment

- Comparison on Other Ensemble Learning Methods

Dataset: UT-Zappos ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

	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 ±0.001	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

Part4 : Experiment

- Comparison on Other Ensemble Learning Methods

Dataset: UT-Zappos ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

	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

Part4 : Experiment

- Comparison on Other Ensemble Learning Methods

Dataset: MIT-States

ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

	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

Part4 : Experiment

- Comparison on Other Ensemble Learning Methods

Dataset: MIT-States

ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

	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

Experimental Results

- Based on these experimental results, it can be observed that **SPE** generally outperforms other methods in terms of performance, while **HMOE-SOE** has an advantage over experts 12 and 13 in the UT-Zappos dataset.

SPE Performance Analysis

- We will investigate the advantages of **SPE** by using the **Variance of Average Error Rate** to observe under which circumstances **SPE** performs better.

Part4 : Experiment

- SPE Performance Analysis

	專家1錯誤率	專家2錯誤率	專家3錯誤率	SPE選擇結果	真實結果	SPE結果是否正確
樣本1	0.45052632689476013	0.3759123980998993	0.37362638115882874	6	6	True
樣本3	0.5949764251708984	0.5875968933105469	0.6447761058807373	14	14	True
樣本7	0.3744075894355774	0.3759123980998993	0.1666666716337204	13	13	True
樣本8	0.45052632689476013	0.4794520437717438	0.52304607629776	6	6	True
樣本11	0.45052632689476013	0.3759123980998993	0.37362638115882874	6	6	True
樣本12	0.3744075894355774	0.3759123980998993	0.1666666716337204	13	13	True
樣本13	0.1818181872367859	0.0833333358168602	0.1666666716337204	13	13	True
樣本15	0.5949764251708984	0.5875968933105469	0.5873016119003296	6	6	True
樣本17	0.5042372941970825	0.5875968933105469	0.52304607629776	6	6	True
樣本18	0.7982261776924133	0.6676300764083862	0.7017208337783813	6	6	True
樣本20	0.5042372941970825	0.5875968933105469	0.5873016119003296	9	9	True
樣本21	0.5042372941970825	0.7146171927452087	0.7017208337783813	13	13	True
樣本24	0.45052632689476013	0.5875968933105469	0.5873016119003296	9	9	True
樣本26	0.45052632689476013	0.5875968933105469	0.6447761058807373	13	13	True
樣本28	0.3744075894355774	0.4794520437717438	0.52304607629776	6	6	True
樣本30	0.45052632689476013	0.5875968933105469	0.6447761058807373	13	13	True
樣本31	0.45052632689476013	0.5875968933105469	0.5873016119003296	9	9	True

Part4 : Experiment

- SPE Performance Analysis

		專家1錯誤率	專家2錯誤率	專家3錯誤率	SPE選擇結果	真實結果	SPE結果是否正確
1	樣本52	0.7136563658714294	0.7146171927452087	0.7017208337783813	13	0	False
2	樣本58	0.5949764251708984	0.5875968933105469	0.5873016119003296	13	3	False
3	樣本73	0.5949764251708984	0.5875968933105469	0.5873016119003296	14	9	False
4	樣本78	0.5949764251708984	0.5875968933105469	0.5873016119003296	13	3	False
5	樣本82	0.5949764251708984	0.5875968933105469	0.5873016119003296	13	3	False
6	樣本87	0.3744075894355774	0.3759123980998993	0.37362638115882874	6	4	False
7	樣本98	0.7136563658714294	0.7146171927452087	0.7017208337783813	14	13	False
8	樣本109	0.7136563658714294	0.7146171927452087	0.7017208337783813	6	4	False
9	樣本136	0.7136563658714294	0.7146171927452087	0.7017208337783813	6	13	False
10	樣本140	0.3744075894355774	0.3759123980998993	0.37362638115882874	6	7	False
11	樣本145	0.3744075894355774	0.3759123980998993	0.37362638115882874	0	14	False
12	樣本146	0.3744075894355774	0.3759123980998993	0.37362638115882874	13	7	False
13	樣本149	0.7136563658714294	0.7146171927452087	0.7017208337783813	6	13	False
14	樣本156	0.7136563658714294	0.7146171927452087	0.7017208337783813	6	14	False
15	樣本171	0.7136563658714294	0.7146171927452087	0.6973365545272827	14	13	False
16	樣本223	0.3744075894355774	0.3759123980998993	0.37362638115882874	6	7	False

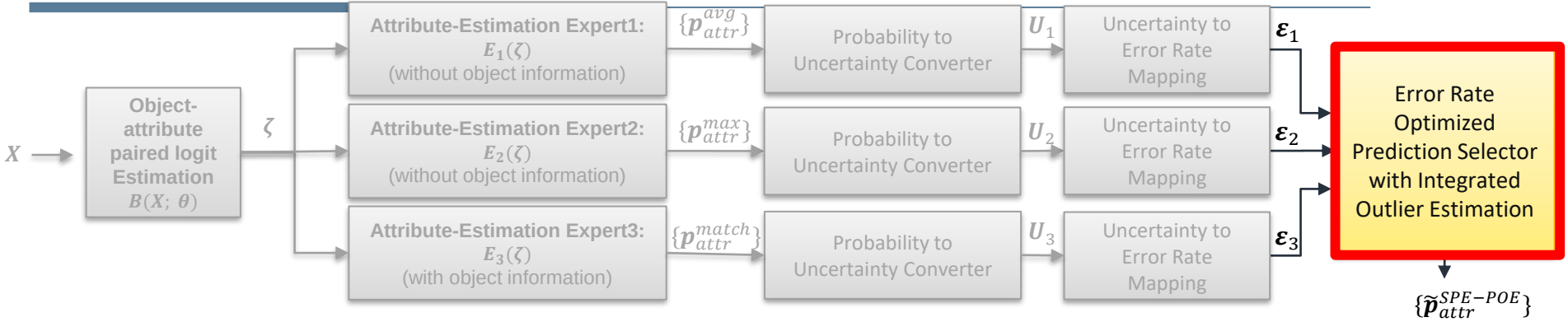
Part4 : Experiment

- SPE Performance Analysis

Previous studies have shown that the variance of the average error rate for correctly predicted samples by SPE is generally high. This suggests that our SPE method can more accurately predict results when there is a high variance in the prediction error rate.

	Variance of average error rate of SPE (correctly predicted samples)	Variance of average error rate of SPE (incorrectly predicted samples)
UT-Zappos	0.0046 ±0.001	0.0039 ±0.001
MIT-States	0.0268 ±0.001	0.0160 ±0.0008

Part3 : Experiment
- Ablation Study: SPE Outlier Estimation via POE Implementation



In our study, we replaced the SOE component within the SPE Outlier Estimation framework with POE, aiming to investigate the impact of this algorithmic substitution on SPE performance.

- **Expert Probability Selection Based on Error Rate Thresholding**

P_{j_i} and P_{k_i} and P_{l_i} are the predicted probabilities for the i-th sample from experts j , k and l respectively.

$$\{\tilde{\boldsymbol{p}}_{attr}^{SPE-\mathrm{POE}}\} = \begin{cases} P_{j_i} * P_{k_i} * P_{l_i} & if\ three\ experts\ j,k,l\ satisfy\ \Delta\boldsymbol{\varepsilon}_{j_i} < \theta\ and\ \Delta\boldsymbol{\varepsilon}_{k_i} < \theta\ and\ \Delta\boldsymbol{\varepsilon}_{l_i} < \theta \\ P_{j_i} * P_{k_i} & if\ two\ experts\ j,k\ satisfy\ \Delta\varepsilon_{j_i} < \theta \text{ and } \Delta\varepsilon_{k_i} < \theta \\ P_{j_i} & if\ one\ expert\ j\ satisfies\ \Delta\boldsymbol{\varepsilon}_{j_i} < \theta \end{cases}$$

Part4 : Experiment

- Ablation Study: SPE Outlier Estimation via POE Implementation

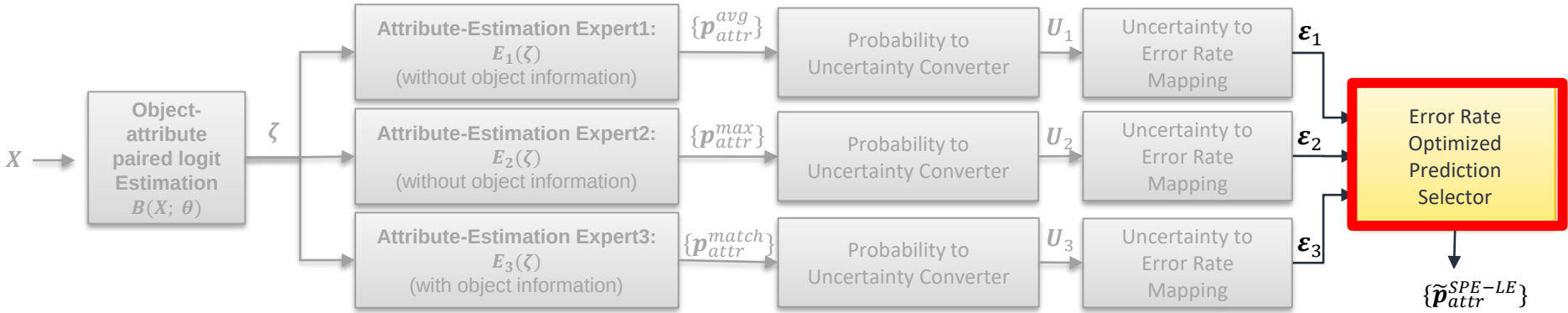
After analyzing the experimental data, we observed a slight performance decline in the SPE-POE framework compared to the original SPE framework. This finding indicates that the effectiveness of using SOE within SPE is superior to that of POE.

Dataset: UT-Zappos	SPE_{SOE}	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

Dataset: MIT-States	SPE_{SOE}	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

Part4 : Experiment

- Ablation study: SPE chooses the lowest error rate



- **Final Prediction Generation:**

The function generates final predictions for each sample based on the selected expert.

$$\tilde{\boldsymbol{p}}_{attr}^{\mathrm{SPE}-LE} = \begin{cases} \boldsymbol{p}_{attr}^{avg} & if\,(\varepsilon_1 < \varepsilon_2) \cap (\varepsilon_1 < \varepsilon_3) \\ \boldsymbol{p}_{attr}^{max} & if\,(\varepsilon_2 < \varepsilon_1) \cap (\varepsilon_2 < \varepsilon_3) \\ \boldsymbol{p}_{attr}^{match} & if\,(\varepsilon_3 < \varepsilon_1) \cap (\varepsilon_3 < \varepsilon_2) \end{cases}$$

Part4 : Experiment

- Ablation study: SPE chooses the lowest error rate

Compared to SPE-LE, SPE shows better overall performance on two datasets. SPE only has some advantages in the UT-Zappos dataset when using the F1 score.

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

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

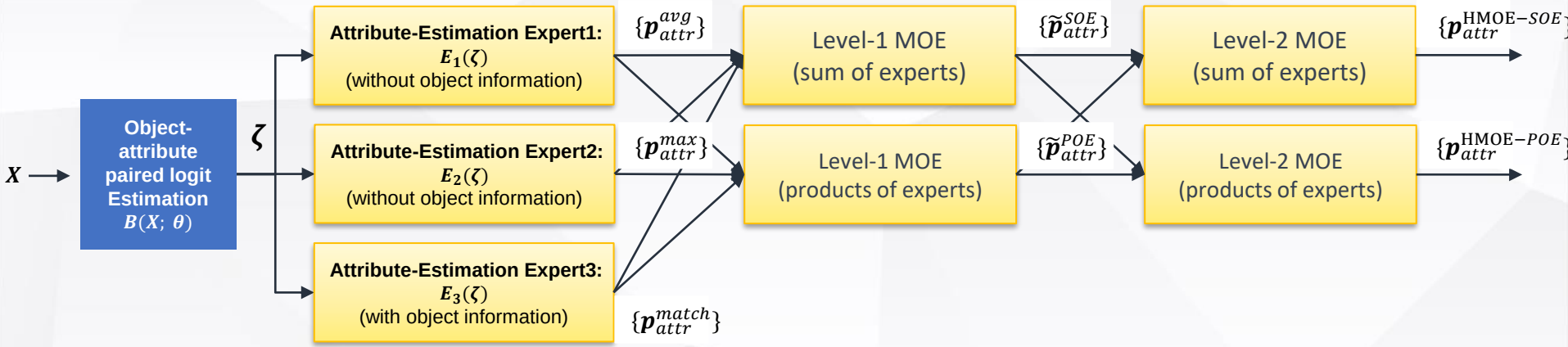
Ablation study: **Hierarchical MOE(HMOE)**

- The purpose of this research is to conduct an ablation study by removing the proposed SPE method to investigate its Effect on accuracy.

- The experiment aims to assess the significance of our method and its influence on the overall research.

Part3 : Method

- Ablation study: Hierarchical MOE(HMOE)



- The method SPE was removed, as shown in the diagram above. This approach allows for an assessment of the Effect of the method on accuracy.

Part4 : Experiment

- Ablation study: Hierarchical mixture of experts (HMOE)

ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

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

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

Part4 : Experiment

- The Effect of Model Calibration

Given the issue of overconfidence in our model, we can utilize the Uncertainty-to-Error-rate Lookup Table through our method-SPE to determine the error rate of each expert. By selecting the prediction with the lowest error rate, we can address the problem of overconfidence in the model.

The performance of our method SPE will be better than the performance of normal model calibration, although the model is overconfident without having to do model calibration to correct it. In the end, we will further use HMOE to integrate these predictions.

The following two experiments demonstrate that uncertainty error rate mapping is effective for model calibration compared to other model calibration methods.

- Using Monte Carlo Dropout to Calibrate Models
- Using Temperature scaling to Calibrate Models

Using Monte Carlo Dropout to Calibrate Models

- This study implements Monte Carlo Dropout on the Logit Estimation Model to calibrate the model and explore the Effect of on the method.

- Experiments will be conducted using Monte Carlo Dropout on two classic datasets of CZSL.

Part4 : Experiment

-Using Monte Carlo Dropout to Calibrate Models on UT-Zappos

Dataset: UT-Zappos ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : Accuracy Model Calibration Method:Monte Carlo Dropout

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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	0.506±0.020	0.5041±0.019

Part4 : Experiment

- Using Monte Carlo Dropout to Calibrate Models on MIT-States

Dataset: MIT-States ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : Accuracy Model Calibration Method: Monte Carlo Dropout

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Using Temperature scaling to Calibrate Models

- The Logit Estimation Model's temperature scaling will be adjusted to calibrate the model, and the effect of calibration on the method will be investigated.

- Two classic datasets of CZSL will be used for experimentation.

Part4 : Experiment

-Using Temperature scaling to Calibrate Models on UT-Zappos

Temperature scaling calibrates models to align prediction uncertainties with ground-truth uncertainties and improve lookup table performance.

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				0.0474 ±0.002	0.3036 ±0.001	0.0665 ±0.011
UT-Zappos Temperature=1.2				**0.0394 ±0.001**	**0.2693 ±0.001**	**0.0306 ±0.001**
UT-Zappos Temperature=1.3				0.0667 ±0.003	0.2363 ±0.002	0.0411 ±0.001

Part4 : Experiment

-Using Temperature scaling to Calibrate Models on UT-Zappos

Dataset: UT-Zappos

ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : Accuracy Model Calibration Method: Temperature scaling

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Part4 : Experiment

-Using Temperature scaling to Calibrate Models on MIT-States

Temperature scaling calibrates models to align predictive uncertainties with real outcomes and enhance lookup table performance.

$ep1 = \text{expert-avg}, \quad ep2 = \text{expert-max}, \quad ep3 = \text{expert-match}$

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				0.022 ±0.0008	0.3514 ±0.001	0.3852 ±0.0009
MIT-State Temperature=1.3				0.0205 ±0.001	0.3448 ±0.0009	0.3689 ±0.001
MIT-State Temperature=1.4				**0.0511 ±0.001**	**0.3415 ±0.001**	**0.3543 ±0.0007**

Part4 : Experiment

-Using Temperature scaling to Calibrate Models on MIT-States

Dataset: MIT-States

ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : Accuracy Model Calibration Method: Temperature scaling

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Experimental Results

Based on these experimental results, we found that SPE performs better on both datasets. After model calibration, the performance of SPE, along with SOE and POE, slightly decreases.

In the UT-Zappos and MIT-States datasets, SPE can replace SOE and POE, and SPE without model calibration outperforms SOE and POE.

Does the model need calibration?
On both datasets, there is no need for model calibration, and the performance of the model declines after calibration.

Conclusion

PART FIVE

Part5: Conclusion

- We proposed two distinct levels of Ensemble learning methods: **SPE** (Selected Prime Expert) and **HMOE** (Hierarchical MOE). **SPE** allows for a more precise identification of which experts exhibit higher confidence and accuracy in specific circumstances. This not only assists in selecting and integrating the most confident prediction results but also further enhances the overall predictive performance.

- **HMOE** involves integrating probabilities output by different Ensemble learning methods. However, the experimental results suggest that using only the SOE and POE methods for integration learning within **HMOE** is inadequate. Our proposed **SPE** method must be incorporated. Thus, the final design of **HMOE** consists of a two-layered MOE. MOE refers to the integration of learning outputs from three experts using the SOE, POE, and **SPE** methods.

- our approach achieved a performance improvement from 51% to 54% on the UT-Zappos dataset and from 29% to 31% on the MIT-States dataset. Furthermore, our experiments were averaged over three seeds to ensure the reliability of the data.

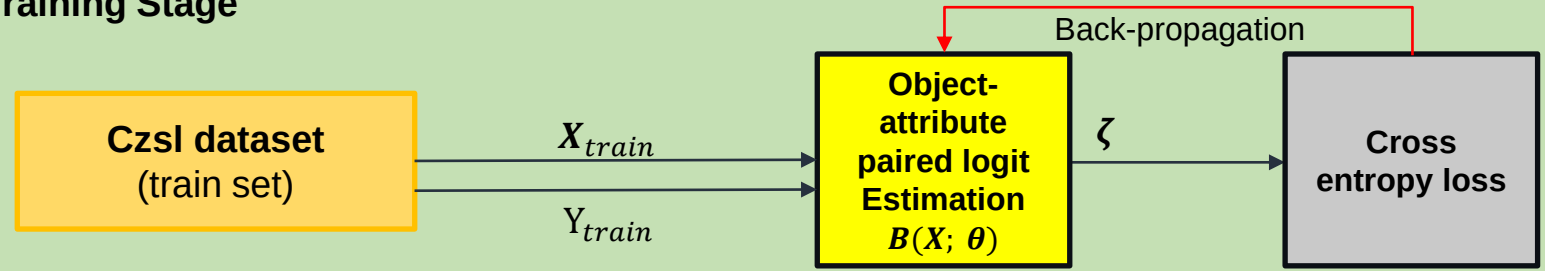
Thanks for your attention

Reporter：周峻緯　　Date：2024/03/27

Appendix.

- Training and Selection of the Object-attribute paired logit Estimation

Training Stage



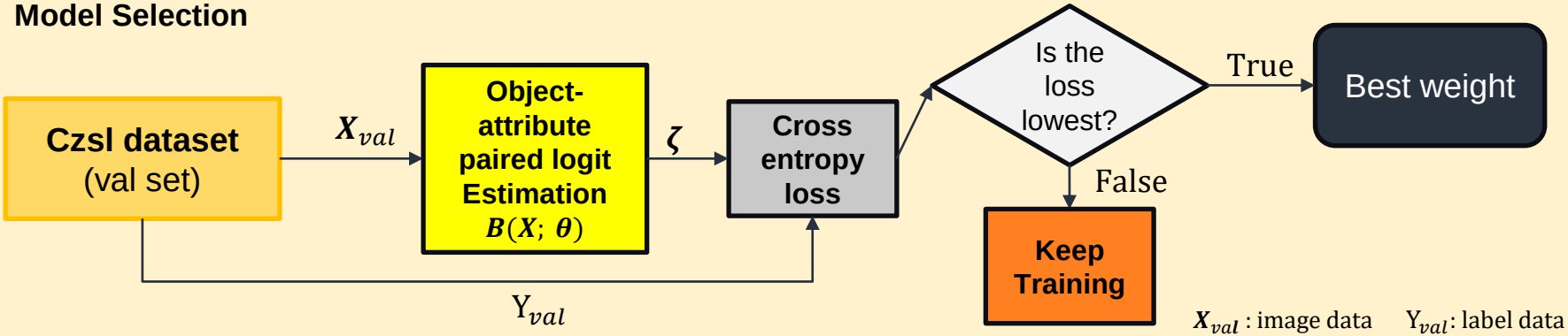
Why do we need a Object-attribute paired logit Estimation? We have pair estimation first, then partial estimation, so we need Object-attribute paired logit Estimation. Object-attribute paired logit Estimation outputs pair estimation.

Object-attribute paired logit Estimation: We've selected the Decomposed Fusion with Soft Prompt(DFSP) model as our baseline for comparison.

ζ : Pairs logits

X_{train} : image data Y_{train}: label data

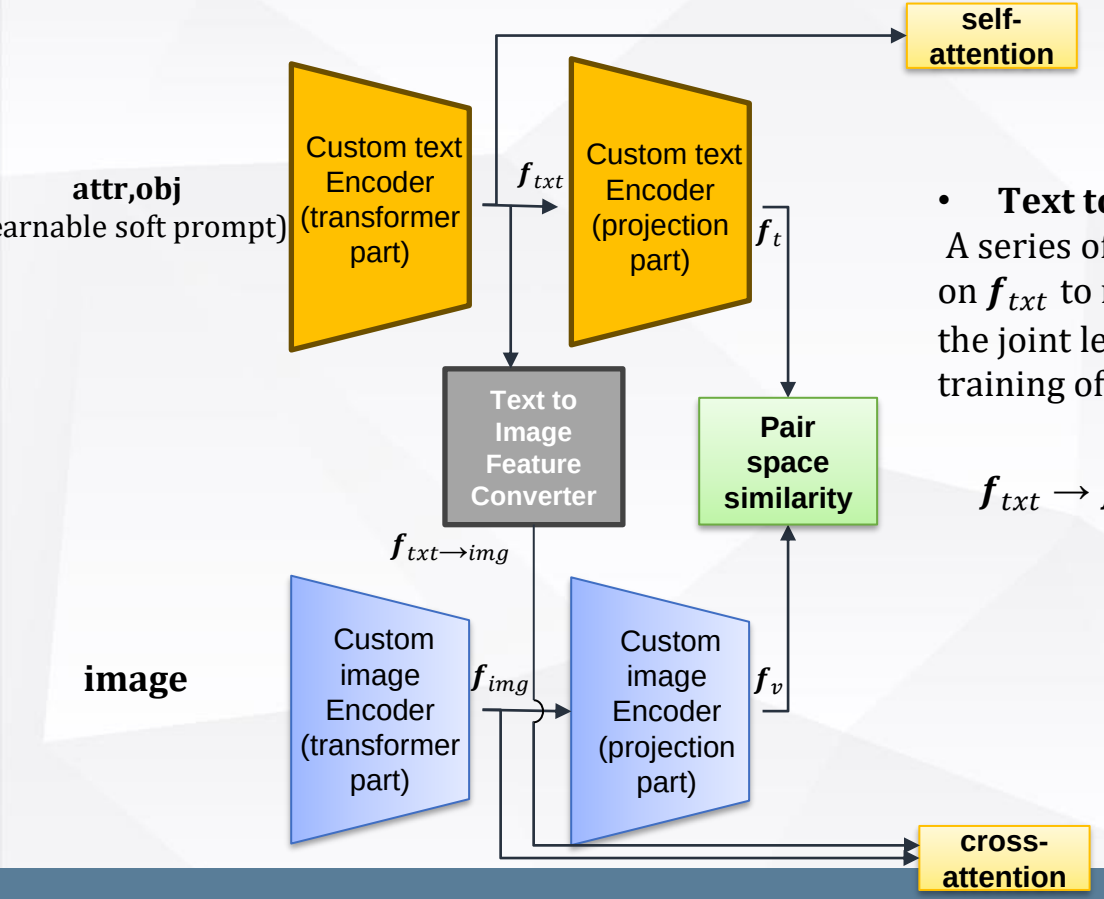
Model Selection



X_{val} : image data Y_{val}: label data

Appendix.

- Logit Estimation Model Architecture: Text to Image Feature Converter

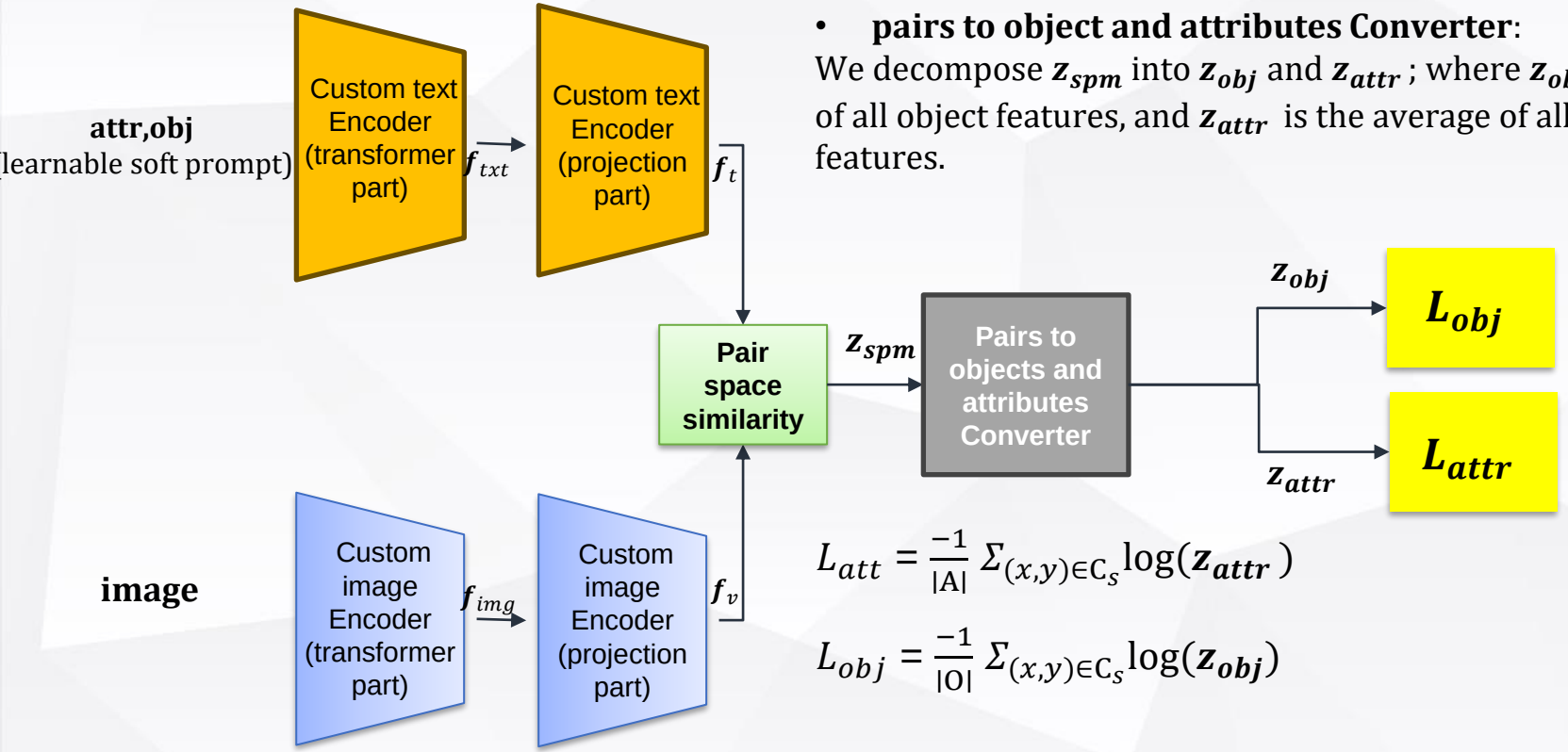


- **Text to Image Feature Converter**:

A series of dimension transformations will be performed on f_{txt} to make it consistent with f_{img} . This will enable the joint learning of text and image features during the training of the Custom Image Encoder.

$$f_{txt} \rightarrow f_{txt \to img}$$

Appendix.
- Logit Estimation Model Architecture: Loss function1



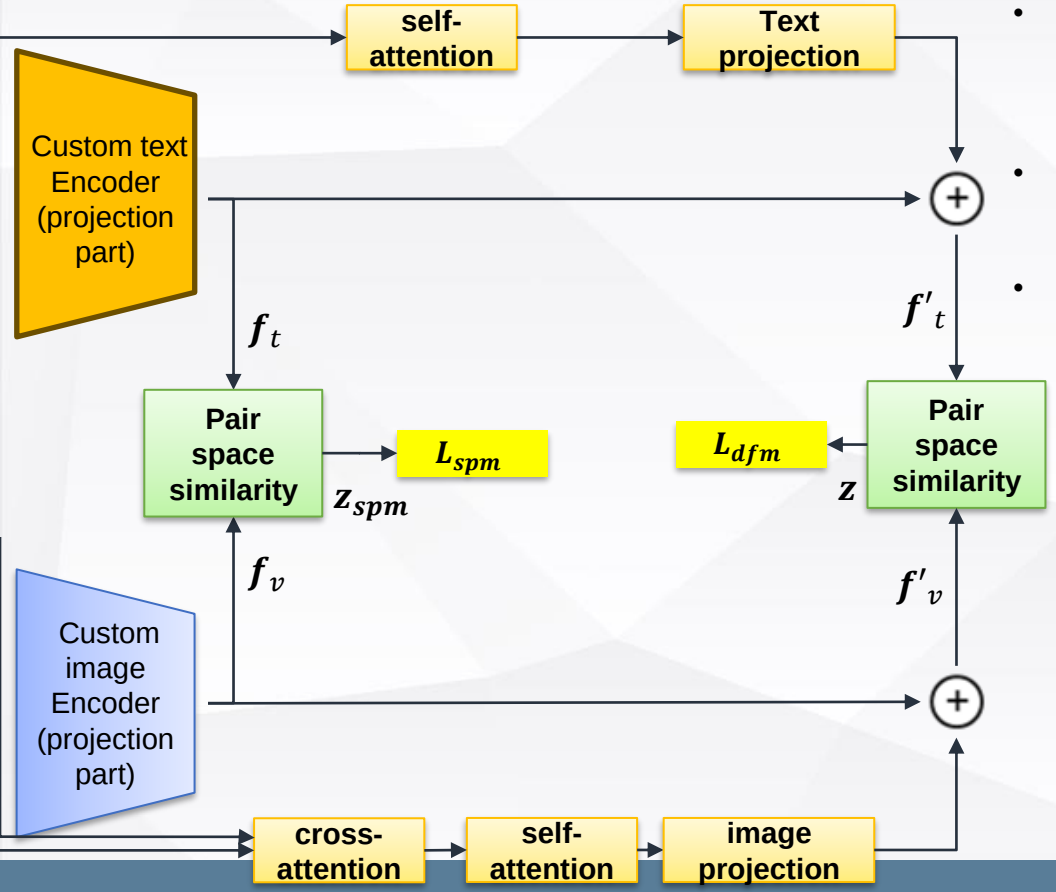
- **pairs to object and attributes Converter**:

We decompose z_{spm} into z_{obj} and z_{attr} ; where z_{obj} is the average of all object features, and z_{attr} is the average of all attribute features.

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

$$L_{obj} = \frac{-1}{|O|} \Sigma_{(x,y)\in C_s} \log(z_{obj})$$

Appendix.
- Logit Estimation Model Architecture: Loss function2



- z_{spm} : Calculate the dot product of the output f_t from the Custom Text Encoder (projection part) and the output f_v from the Custom Image Encoder (projection part) to obtain the model output logit z_{spm}.
- $\mathbf{z}$: Calculate the dot product of the fused text feature f_t' and the fused image feature f_v' to obtain the model output logit $\mathbf{z}$.
- $\mathbf{L_{spm}}$ and $\mathbf{L_{dfm}}$:Use the two model outputs, $\mathbf{z}_{spm}$ and $\mathbf{z}$, to calculate L_{spm} and L_{dfm} respectively.

$$z_{spm}\left(\frac{y=(s,o)}{x:\theta}\right)=\frac{\exp(f_v\cdot f_t)}{\Sigma_{(\bar{o},\bar{s})\in C_s\cup C_{u\_}}\exp(f_v\cdot f_t)}.$$

$$z\left(\frac{y=(s,o)}{x:\theta}\right)=\frac{\exp(f'_v\cdot f'_t)}{\Sigma_{(\bar{o},\bar{s})\in C_s\cup C_{u\_}}\exp(f'_v\cdot f'_t)}.$$

$$L_{spm}=\frac{-1}{|C_s|}\Sigma_{(x,y)\in C_s}\log\left(z_{spm}(\frac{y=(s)}{x;\theta})\right)$$

$$L_{dfm}=\frac{-1}{|C_s|}\Sigma_{(x,y)\in C_s}\log\left(z(\frac{y=(s)}{x;\theta})\right)$$

Appendix.
- Evaluation Metrics - uncertainty calibration error

We use normalized entropy as a measure of uncertainty and introduce the Uncertainty Calibration Error (UCE), a calibration metric for multi-class classification. UCE avoids the pitfalls of other metrics while maintaining high interpretability.

$$\mathrm{UCE} = \sum_{m=1}^{M} \frac{|\boldsymbol{B}_m|}{n} |\boldsymbol{\varepsilon}(\boldsymbol{B}_m) - U(\boldsymbol{B}_m)|$$

$$\boldsymbol{\varepsilon}(B_m) := \frac{1}{|\boldsymbol{B}_m|} \sum_{i \in \boldsymbol{B}_m} 1\,(\hat{\boldsymbol{y}}_i \neq \boldsymbol{y})$$

$$U(\boldsymbol{B}_m) := \frac{-1}{|\boldsymbol{B}_m|} \sum_{i \in B_m} H(\boldsymbol{p}_i).$$

$$H(\boldsymbol{p}_i) := \frac{1}{log\ \mathrm{C}} \sum_{c=1}^{C} -\boldsymbol{p}^{(c)} log\ \boldsymbol{p}^{(c)},\ H(\boldsymbol{p}_i) \in [0,1]\ .$$

Appendix.

-Using Monte Carlo Dropout to Calibrate Models on UT-Zappos

Dataset: UT-Zappos ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : F1 score Model Calibration Method:Monte Carlo Dropout

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Appendix.

-Using Monte Carlo Dropout to Calibrate Models on MIT-States

Dataset: MIT-States ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : F1 score Model Calibration Method:Monte Carlo Dropout

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Appendix.

-Using Monte Carlo Dropout to Calibrate Models on UT-Zappos

Dataset: UT-Zappos ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : F1 score Model Calibration Method: Temperature scaling

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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

Appendix.

-Using Monte Carlo Dropout to Calibrate Models on MIT-States

Dataset: MIT-States ep1 = expert-avg, ep2 = expert-max , ep3 = expert-match

Evaluation Metrics : F1 score Model Calibration Method: Temperature scaling

	SPE (Without Model calibration)	SOE (Without Model calibration)	POE (Without Model calibration)	SPE (Model Calibration)	SOE (Model Calibration)	POE (Model Calibration)
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