On the Calibration of Learning to Defer to Multiple Experts

Rajeev Verma [1] **Daniel Barrejón** [2] **Eric Nalisnick** [1]

Abstract

We study the calibration properties of multi-expert *learning to defer* (L2D). In particular, we study the framework's ability to estimate $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$, the probability that the jth expert will correctly predict the label for $\boldsymbol{x}$. We compare softmax- and one-vs-all-parameterized L2D, finding the former causes mis-calibration to propagate between the estimates of expert correctness while the latter's parameterization does not.

1. Introduction

In human-machine collaboration, the primary challenge is often thought to be when to rely on the machine vs the human. Yet when there are multiple experts, there are two decisions to be made: *when* to defer and *to whom* to defer. Some experts may perform better than the model but perhaps others will not. Thus assessing and monitoring expert quality is an important sub-task.

In this work, we analyze the forecasting properties of a hybrid intelligence system (Kamar, 2016) involving multiple experts (Keswani et al., 2021). In particular, we study the calibration properties of multi-expert *learning to defer* (Madras et al., 2018). We compare the softmax-based (Mozannar & Sontag, 2020) and one-vs-all-based (Verma & Nalisnick, 2022) formulations, finding that due to the former's tied parameterization, calibration error can propagate across experts. The one-vs-all parameterization does not have this behavior due to its independence assumptions. We perform experiments on simulated (mixture of Gaussians) and real (CIFAR-10) data showcasing the consequence of this difference. Ultimately, we find that the softmax parameterization becomes increasingly poorly calibrated as the number of experts in the system increases.

2. Learning To Defer to Multiple Experts

Data We first define the data for multi-class, multi-expert *learning to defer* (L2D). Let $\mathcal{X}$ denote the feature space, and let $\mathcal{Y}$ denote the output space, which we will always assume to be a categorical encoding of multiple (K) classes. We assume that we have samples from the true generative process: $\mathbf{x}_n \in \mathcal{X}$ denotes a feature vector, and $\mathrm{y}_n \in \mathcal{Y}$ denotes the associated class defined by $\mathcal{Y}$ (1 of K). The L2D problem also assumes that we have access to (usually human) expert demonstrations. Let there be J experts, and denote each expert's prediction space as $\mathcal{M}_j$, which is usually taken to be equal to the label space: $\mathcal{M}_j = \mathcal{Y}$. The expert demonstrations are denoted $\mathrm{m}_{n,j} \in \mathcal{M}_j$ for the associated features $\mathbf{x}_n$. The combined N-element training sample is $\mathcal{D} = \{\boldsymbol{x}_n, y_n, m_{n,1}, \ldots, m_{n,J}\}_{n=1}^{N}$.

Models The L2D framework is built from the classifier-rejector approach (Cortes et al., 2016a;b). The goal is to learn two functions: the *classifier*, $h : \mathcal{X} \to \mathcal{Y}$, and the *rejector*. In L2D with one expert, the rejector makes a binary decision—to defer or not—but in multi-expert L2D, the rejector also must choose to which expert to assign the instance. Let the rejector be denoted $r : \mathcal{X} \to \{0, 1, \ldots, J\}$. When $r(\mathbf{x}) = 0$, the classifier makes the decision in the typical way. When $r(\mathbf{x}) = j$, the classifier abstains and defers the decision to the jth expert.

Softmax Surrogate Loss The learning problem requires fitting both the rejector and classifier. Mozannar & Sontag (2020) proposed the first consistent surrogate loss for L2D. They accomplish this by first unifying the classifier and rejector via an augmented label space that includes the rejection option. Formally, this label space is defined as $\mathcal{Y}^{\perp} = \mathcal{Y} \cup \{\perp_1, \ldots, \perp_J\}$ where $\perp_j$ denotes the decision to defer to the jth expert. Secondly, Mozannar & Sontag (2020) use a reduction to cost sensitive learning that ultimately resembles the cross-entropy loss for a softmax parameterization.

While the multi-expert setting was not investigated by Mozannar & Sontag (2020), it is straightforward to extend their formulation to include J experts. Let the classifier be composed of K functions: $g_k : \mathcal{X} \mapsto \mathbb{R}$ for $k \in [1, K]$ where k denotes the class index. The rejector is implemented with J functions: $g_{\perp,j} : \mathcal{X} \mapsto \mathbb{R}$ for $j \in [1, J]$

[1] Informatics Institute, University of Amsterdam, Amsterdam, Netherlands [2] Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Madrid, Spain. Correspondence to: Rajeev Verma <rajeev.ee15@gmail.com>.

ICML 2022 Workshop on Human-Machine Collaboration and Teaming. Copyright 2022 by the author(s).

where j is the expert index. These $K + J$ functions are then combined via the following softmax-parameterized loss:

$$\phi_{\text{SM}}\left(g_1, \ldots, g_K, g_{\perp,1}, \ldots, g_{\perp,J}; \boldsymbol{x}, y, m_1, \ldots, m_J\right) = \\ -\log\left(\frac{\exp\{g_y(\boldsymbol{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\boldsymbol{x})\}}\right) \\ -\sum_{j=1}^{J} \mathbb{I}[m_j = y] \, \log\left(\frac{\exp\{g_{\perp,j}(\boldsymbol{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\boldsymbol{x})\}}\right).$$

The intuition is that the first term maximizes the function g_k associated with the true label. The second term maximizes the rejection function $g_{\perp,j}$ but only if the jth expert's prediction is correct. At test time, the classifier is obtained by taking the maximum over $k \in [1, K]$: $\hat{y} = h(\boldsymbol{x}) = \arg\max_{k \in [1,K]} g_k(\boldsymbol{x})$. The rejection function is similarly formulated as

$$r(\boldsymbol{x}) = \begin{cases} 0 \quad \text{if} \quad g_{h(\boldsymbol{x})} > g_{\perp,j'} \quad \forall j' \in [1, J] \\ \arg\max_{j \in [1,J]} g_{\perp,j}(\boldsymbol{x}) \quad \text{otherwise.} \end{cases}$$

Using the same proof techniques as for the single expert setting, ϕ_{SM} is shown to be a convex (in g) and consistent surrogate loss for $0-1$-cost multi-expert L2D. The minimizers $g_1^*, \ldots, g_K^*, g_{\perp,1}^*, \ldots, g_{\perp,J}^*$ result in the optimal classifier and rejector, satisfying:

$$h^*(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \; \mathbb{P}(\mathrm{y} = y | \boldsymbol{x}), \\ r^*(\boldsymbol{x}) = \begin{cases} 0 \text{ if } \mathbb{P}(\mathrm{y} = h^*(\boldsymbol{x}) | \boldsymbol{x}) > \mathbb{P}(\mathrm{m}_{j'} = \mathrm{y} | \boldsymbol{x}) \;\; \forall j' \\ \arg\max_{j \in [1,J]} \mathbb{P}(\mathrm{m}_j = \mathrm{y} | \boldsymbol{x}) \quad \text{otherwise,} \end{cases}$$

where $\mathbb{P}(\mathrm{y}|\boldsymbol{x})$ is the probability of the label under the data generating process, and $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$ is the probability that the jth expert is correct.

One-vs-All Surrogate Loss Verma & Nalisnick (2022) proposed an alternative consistent surrogate for L2D based on a one-vs-all (OvA) formulation. It too can be straightforwardly extended to the multi-expert setting:

$$\psi_{\text{OvA}}(g_1, \ldots, g_K, g_{\perp,1}, \ldots, g_{\perp,J}; \boldsymbol{x}, y, m_1, \ldots, m_J) = \\ \phi[g_y(\boldsymbol{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\boldsymbol{x})] \; + \sum_{j=1}^{J} \phi[-g_{\perp,j}(\boldsymbol{x})] \\ + \sum_{j=1}^{J} \mathbb{I}[m_j = y] \left(\phi[g_{\perp,j}(\boldsymbol{x})] - \phi[-g_{\perp,j}(\boldsymbol{x})]\right)$$

where $\phi : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_+$ is a binary surrogate loss. For instance, when ϕ is the logistic loss, we have $\phi[f(\boldsymbol{x})] = \log(1 + \exp\{-f(\boldsymbol{x})\})$. The g-functions are entirely the same, and the classifier and rejector are computed exactly as in the softmax case.

3. Calibration of Expert Confidence

For both types of L2D parameterizations, we are interested in studying the *calibration* of the system (Dawid, 1982). In particular, we are interested in the model's ability to estimate $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$, the probability that the jth expert is correct for a particular instance. If the L2D system says that $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x}_0) = 0.7$, then the jth expert should be correct 70% of the time for inputs very similar to $\boldsymbol{x}_0$. This quantity is crucial not only for the system's ability to correctly defer but is also useful for interpretability and safety—to quantify what the model thinks the human knows. Our study is inspired by the work of Verma & Nalisnick (2022), who found that the Mozannar & Sontag (2020) parameterization can result in poor estimators in practice, despite having valid Bayes optimal solutions. We wish to examine each parameterization's behavior in the multi-expert formulation.

3.1. Softmax Parameterization

In the Mozannar & Sontag (2020) formulation, the estimator of the probability that the jth expert is correct can be derived as follows; see Appendix B.1 for a complete derivation. For the Bayes optimal functions $g_1^*, \ldots, g_{\perp,J}^*$, we have:

$$\frac{\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})}{1 + \sum_{j'=1}^{J} \mathbb{P}(\mathrm{m}_{j'} = \mathrm{y}|\boldsymbol{x})} = \underbrace{\frac{\exp\{g_{\perp,j}^*\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}^*(\boldsymbol{x})\}}}_{p_{\perp,j}^*(\boldsymbol{x})}. \tag{1}$$

Denote the RHS of Equation 1 as $p_{\perp,j}^*(\boldsymbol{x})$. Since we have J equations, one for each expert, we can uniquely solve for $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$ as:

$$\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x}) = \frac{p_{\perp,j}^*(\boldsymbol{x})}{1 - \sum_{j'=1}^{J} p_{\perp,j'}^*(\boldsymbol{x})}. \tag{2}$$

Due to the denominator involving the quantity $p_{\perp,j}^*(\boldsymbol{x})$ for all experts, there is dependence across the estimators.

For the single expert softmax parameterization, Verma & Nalisnick (2022) observed that the estimated probability of expert correctness could be degenerate—that is, greater than one. We see the exact same issue in Equation 2: for $p_{\perp,j}(\boldsymbol{x}) > 0$, as $\sum_{j'=1}^{J} p_{\perp,j'}(\boldsymbol{x})$ approaches one, the estimate for $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$ will go to infinity. Of course, the model will no longer be at its Bayes optimal configuration if this degeneracy occurs, so in the experiments we will test if this degeneracy can occur in practice.

3.2. One-vs-All Parameterization

For the OvA formulation (Verma & Nalisnick, 2022), the probability that the expert is correct is directly modeled by the jth deferral function. For the logistic binary loss ϕ, it is given as:

$$\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x}) \;=\; \frac{1}{1+\exp\{-g^*_{\perp,j}(\boldsymbol{x})\}}. \qquad (3)$$

This estimator has the correct range of $(0, 1)$ for any setting of $g_{\perp,j} \in \mathbb{R}$. Moreover, there is no dependence across expert deferral functions $g_{\perp,1}, \ldots, g_{\perp,J}$.

4. Related Work

Calibration has been identified as important to fostering trust in hybrid intelligence systems (Schmidt & Biessmann, 2020; Zhang et al., 2020) While recent work has studied multi-expert rejector-classifier systems (Grønsund & Aanestad, 2020; Keswani et al., 2021), none has examined their calibration properties. Verma & Nalisnick (2022) study calibration in the single-expert L2D setting, which directly motivates this work.

5. Experiments

We perform experiments on synthetic data and on the CIFAR-10 dataset. In both cases, we measure calibration according to the *expected calibration error* (ECE):

$$\mathrm{ECE}(p_{\perp,j}) = \mathbb{E}_{\mathbf{x}}|\mathbb{P}\left(\mathrm{m}_j = \mathrm{y} \mid p_{\perp,j}(\mathbf{x}) = c\right) - c|.$$

Since the softmax parameterization can result is probability estimates greater than one, we cap confidences at 1.0 to calculate ECE in all experiments. First, we study the effect of gradually increasing the number of experts on the overall calibration of the system. Second, we examine how different expert's behavior affects other expert estimates. Our results suggest that systems trained with the softmax surrogate exhibit degradation in calibration as the number of experts increases. Furthermore, other experts in the committee significantly affect the calibration of other experts.

5.1. Datasets and Models

Mixture of Gaussians For the synthetic dataset, we generate a mixture of Gaussians (MoG) with 4 clusters. The data is plotted in Figure 3. It shows severe overlap between cluster 2 and cluster 3, and thus these clusters represent where the expert's advice might be required. The other two clusters have small overlap and can be discriminated by a simple classifier. For the classifier, we use a small feedforward neural network with four layers. We train it using stochastic gradient descent (SGD) with early stopping (look-ahead of 20 epochs).

CIFAR-10 For the experiments using CIFAR-10, we use the canonical train-test split (Krizhevsky, 2009). We partition the training split $90\% - 10\%$ to form training and validation sets, respectively. We use the same neural network and training settings for both the OvA and softmax methods as described in Mozannar & Sontag (2020). We use a wide residual networks (Zagoruyko & Komodakis, 2016) to parameterize the $g(\boldsymbol{x})$ functions. We train a 28-layer network using SGD with momentum and a cosine annealing schedule for the learning rate. We again employ early stopping with 20 look-ahead epochs.

5.2. Effect of Increasing Number of Experts

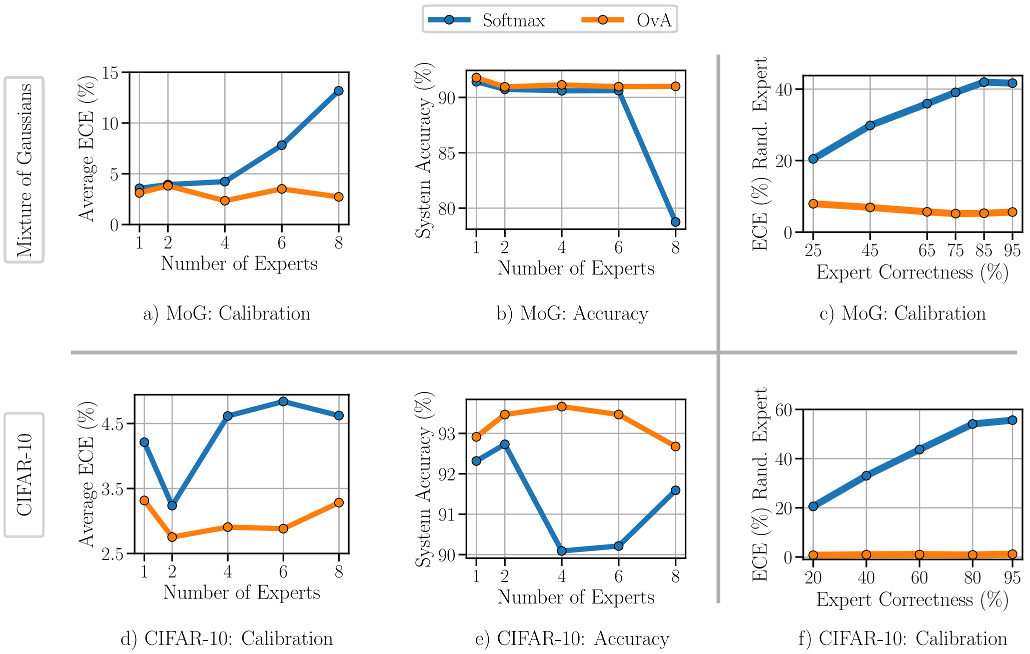
We first examine calibration under an increasing number of experts—from 1 to 8. For the MoG dataset, the experts are oracles if an instance belongs to either cluster 3 or 4 and predict randomly over all classes otherwise. For the CIFAR-10 dataset, the experts are an oracle for the first 5 classes and predict randomly over all 10 classes otherwise.

The results are reported in Figure 1 (a, b, d, e). Firstly, examine subfigures (b) and (e), which report the system accuracy to ensure both parameterizations are well-performing. We see that the OvA parameterization is slightly-to-moderately superior in all cases. Moving on to the calibration results, the ECE is reported in subfigures (a) and (d). We see that the OvA parameterization (orange) is roughly stable w.r.t. expert size, but the softmax (blue) ECE tends to increase. This behavior is expected for the softmax according to Equation 2. With the addition of more experts, the denominator becomes smaller, leading to overconfident (and degenerate) estimates for $\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})$. However, we do see some cancellation effect with the addition of second expert in Figure 1 (d). This can be explained by the fact that adding more experts constrains the confidence allocation to multiple experts (due to the tied nature of the softmax parameterization). But the effect dissipates for 4 or more experts, with the ECEs continuing to increase.

5.3. Expert Dependence

We further aim to assess calibration when there is a gap in expert quality. We simulate four experts with one always being random and the other three having an increasing probability of correctness (20% - 95%). For the MoG dataset, three experts will increase their probability of being correct on two of the clusters and predict randomly for the other two. For CIFAR-10, three experts increase their probability of being correct in the first five classes and predict randomly for the other ten classes. We hypothesize that for the softmax, the calibration of the random expert will increase when the probability of correctness for the other three experts increases due to the tied parameterization. We conjecture that no such dependence will be present in the OvA results.

The results are reported in the third column of Figure 1. We see that the ECE for the random expert dramatically increases for both datasets for the softmax parametrization



a) MoG: Calibration

b) MoG: Accuracy

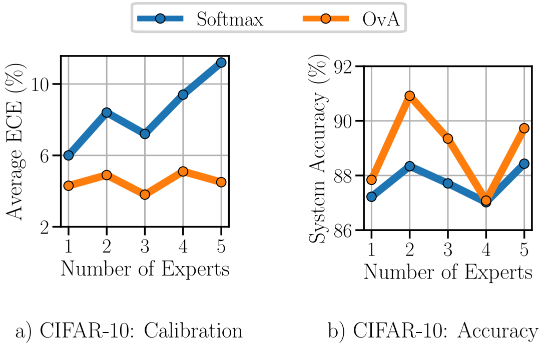
c) MoG: Calibration

d) CIFAR-10: Calibration

e) CIFAR-10: Accuracy

f) CIFAR-10: Calibration

Figure 1. Calibration and System Accuracy on Simulated Data and CIFAR-10. The first column reports ECE under an increasing number of experts, the second column the system accuracy, and the third column the ECE to show the dependence across experts. The top row shows results for the mixture of Gaussians simulation, and the bottom row shows results for CIFAR-10.



a) CIFAR-10: Calibration

b) CIFAR-10: Accuracy

Figure 2. Calibration and System Accuracy on CIFAR-10. Subfigure (a) reports the ECE when an increasing number of specialized experts are incorporated. Subfigure (b) reports the system accuracy under the same conditions.

(blue), reaching values above 40%. Yet for OvA (orange), the ECE is nearly flat for both datasets due its explicit independence across experts. This supports our hypothesis from above that the softmax parameterization will skew per-expert estimation due to its dependencies across experts.

5.4. Specialized Experts

For our final experiment, we examine calibration when the experts have non-overlapping expertise. For CIFAR-10, each expert is simulated to be an oracle on two of the ten classes. Figure 2 reports the average ECE across experts (a) and the system accuracy (b) as the number of specialized experts increases. For system accuracy, both methods are competitive, with OvA (orange) having a slight edge. For ECE, OvA is again clearly superior by being stable across the number of experts. Thus we see that despite the experts having independent expertise, the softmax parameterization still accumulates calibration errors.

6. Conclusions and Future Work

We have demonstrated on simulated and real data that the softmax parameterization of multi-expert L2D exhibits calibration error, especially for an increasing number of experts. The OvA parameterization, on the other hand, is much more stable for multiple experts. We believe that this is explained by the softmax's estimator having dependencies across experts, which causes errors to propagate. Studying multi-expert deferral systems where experts can collaborate, faithfully communicate with each other, etc. remains an open problem. It would also be interesting to study establishing synergies between multiple experts who might have different information and work under different assumptions.

Acknowledgements

This publication is part of the project *Continual Learning under Human Guidance* (VI.Veni.212.203), which is financed by the Dutch Research Council (NWO).

References

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, 2016a.

Cortes, C., DeSalvo, G., and Mohri, M. Boosting with abstention. In *Advances in Neural Information Processing Systems*, 2016b.

Dawid, A. P. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

Grønsund, T. and Aanestad, M. Augmenting the algorithm: Emerging human-in-the-loop work configurations. *The Journal of Strategic Information Systems*, 29(2):101614, 2020.

Kamar, E. Directions in hybrid intelligence: Complementing ai systems with human intelligence. In *International Joint Conference on Artificial Intelligence*, pp. 4070–4073, 2016.

Keswani, V., Lease, M., and Kenthapadi, K. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 154–165, 2021.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Madras, D., Pitassi, T., and Zemel, R. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

Mozannar, H. and Sontag, D. A. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Schmidt, P. and Biessmann, F. Calibrating human-ai collaboration: Impact of risk, ambiguity and transparency on algorithmic bias. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 431–449. Springer, 2020.

Verma, R. and Nalisnick, E. Calibrated learning to defer with one-vs-all classifiers. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *British Machine Vision Conference*, 2016.

Zhang, Y., Liao, Q. V., and Bellamy, R. K. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 295–305, 2020.

A. Mixture of Gaussian Dataset

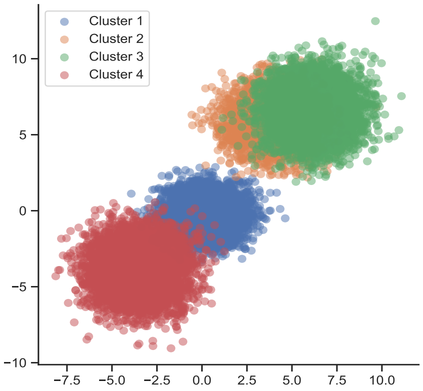


Figure 3. Generated Mixture of Gaussian dataset. The dataset represents varying level of complexity for the simple classifier with cluster 2 and cluster 3 demonstrating severe overlap conducive to querying the expert for correct prediction. The other two clusters are easy to be learned by the classifier.

B. Multi-Expert Learning to Defer: Further Details

The Multi-Expert Learning to Defer setup as studied in this paper is a straightforward extension of Learning to Defer framework proposed and analysed in Mozannar & Sontag (2020) and Verma & Nalisnick (2022). As such the argument for the optimal rejector and classifier of the Multi-Expert Learning to Defer can also be straightforwardly drawn from Proposition 9 of Mozannar & Sontag (2020). According to the proof of Proposition 9 in their paper, we should defer to the expert only if the expected loss of expert making the prediction is less than that of the classifier. This argument can be extended to multi-expert setting that, among all the J experts, the system should compare the expected loss of each of the expert and the classifer, and defer to j^{th} expert if the expected loss of the j^{th} expert is less than expected loss of the classifier. Thus, we get the following Bayes optimal classifier $(h^*(\boldsymbol{x}))$ and rejector $(r^*(\boldsymbol{x}))$:

$$h^*(\boldsymbol{x}) = \arg\max_{y\in\mathcal{Y}} \ \mathbb{P}(\mathrm{y}=y|\boldsymbol{x}),$$
$$r^*(\boldsymbol{x}) = \begin{cases} 0 \text{ if } \mathbb{P}(\mathrm{y}=h^*(\boldsymbol{x})|\boldsymbol{x}) > \mathbb{P}(\mathrm{m}_{j'}=\mathrm{y}|\boldsymbol{x}) \ \ \forall j' \\ \arg\max_{j\in[1,J]} \mathbb{P}(\mathrm{m}_j=\mathrm{y}|\boldsymbol{x}) \quad \text{otherwise,} \end{cases}$$

where $\mathbb{P}(\mathrm{y}|\boldsymbol{x})$ is the probability of the label under the data generating process, and $\mathbb{P}(\mathrm{m}_j=\mathrm{y}|\boldsymbol{x})$ is the probability that the jth expert is correct.

With the above Bayes optimal classifier and the rejector for Multi-Expert Learning to Defer setting, it is again straightforward to use the argument from Mozannar & Sontag (2020) to conclude that ϕ_{SM} is a consistent surrogate loss for multi-expert Learning to Defer. Similar things can be said for ψ_{OvA} based on Verma & Nalisnick (2022).

B.1. Derivation of Eq. 1

The derivation easily follows from the proof of Theorem 2 of Mozannar & Sontag (2020). We follow their proof to write the risk, denoted as $L\left(g_1,\ldots,g_{\perp,J};\boldsymbol{x},y,m_1,\ldots,m_J\right)$, for multi-expert learning to defer as follows:

$$L\left(g_1,\ldots,g_{\perp,J};\boldsymbol{x},y,m_1,\ldots,m_J\right) =$$
$$-\sum_{y\in\mathcal{Y}} \eta_y(\boldsymbol{x}) \log\left(\frac{\exp\{g_y(\boldsymbol{x})\}}{\sum_{y'\in\mathcal{Y}^\perp}\exp\{g_{y'}(\boldsymbol{x})\}}\right) - \sum_{j=1}^{J} \mathbb{P}(\mathrm{m}_j=\mathrm{y}|\boldsymbol{x}) \log\left(\frac{\exp\{g_{\perp,j}(\boldsymbol{x})\}}{\sum_{y'\in\mathcal{Y}^\perp}\exp\{g_{y'}(\boldsymbol{x})\}}\right) \tag{4}$$

We take the partial derivatives with respect to each g function and set them to 0. Placing in the optimal classifier h^*, and taking derivative with respect to g_j and setting it to 0, we get the desired relationship for the optimal g_j^*:

$$\frac{\mathbb{P}(\mathrm{m}_j = \mathrm{y}|\boldsymbol{x})}{1 + \sum_{j'=1}^{J} \mathbb{P}(\mathrm{m}_{j'} = \mathrm{y}|\boldsymbol{x})} \;=\; \frac{\exp\{g^*_{\perp,j}\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g^*_{y'}(\boldsymbol{x})\}}. \tag{5}$$