Methods for Calibrated Uncertainty Quantification and Understanding its Utility

by

Youngseog Chung

Thesis Proposal
Department of Machine Learning
Carnegie Mellon University
2025

Doctoral Committee:

Jeff Schneider (Chair)
Aarti Singh
Zico Kolter
Jasper Snoek (Google DeepMind)

Youngseog Chung
youngsec@cs.cmu.edu
ORCID iD:

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

As machine learning models have become more capable of dealing with complex data, they have been entrusted with an increasing array of predictive tasks. However, with growing reliance on model predictions, being able to assess whether a given model prediction is reliable has become equally important. Uncertainty quantification (UQ) plays a critical role in this context by providing a measure of confidence in a model's predictions, and the quantified uncertainty is considered correct if it is calibrated. In this proposal, I address the problem of optimizing for calibration, especially with regression models which output a distribution over continuous-valued outputs. In my initial work, I propose a collection of methods and techniques to train a quantile model end-to-end with differentiable loss functions that optimize directly for the calibration of the predictive quantiles. This works falls under a class of pre-hoc methods which aim to improve calibration during the training of the model and distinguishes itself from the relatively richer line of work in post-hoc calibration, which aim to calibrate a pre-trained predictive model. Afterwards, I introduce a method to feasibly extend the notion of calibration to multi-dimensional distributions and describe a post-hoc calibration (or recalibration) algorithm. I further discuss how distributional predictions are utilized in applications such as decision-making tasks or model-based reinforcement learning and point out that each application setting requires different qualities for the distributional prediction. In light of this observation, I propose several research directions which study applications of *using* distributional predictions. In particular, I propose re-investigating proper scoring rules as a tool for eliciting good/useful behavior from distributional predictions in a pre-hoc manner.

CHAPTER 1

Introduction

As machine learning models have become more capable of dealing with complex data, they have been entrusted with an increasing array of predictive tasks. These tasks range anywhere from predicting the trajectory of vehicles for autonomous driving [Galvão and Huda, 2024], predicting the structure of protein [Jumper et al., 2021], and even modeling the dynamics of plasma during nuclear fusion reactions [Char et al., 2024]. With growing reliance on model predictions for increasingly complex tasks, one important question a user of these models may ask is whether these model can be trusted. Uncertainty quantification (UQ) plays a critical role in this context by providing a measure of confidence in a model's predictions. Given that most machine learning models are designed and trained in a probabilistic manner, we can leverage the predicted probabilities as a means to express uncertainty: the model can output highly diffuse probabilities when it is uncertain, and concentrated probabilities when it is confident.

When these predicted probabilities align with the frequency of true outcomes, the probabilities are said to be *calibrated*. Calibration is an interpretable metric for UQ which measures the alignment between predicted probabilities and empirical frequencies. This thesis will focus on eliciting calibrated probabilities from machine learning models and further investigate its utility in downstream applications.

Overview of Proposal In the first section of this proposal, we will highlight prior work on optimizing calibration of univariate probabilistic regression models. Afterwards in the second section, we will discuss additional prior work which extend calibration to the multivariate regression setting. The multivariate setting necessitates alternate definitions of calibration, and correspondingly new methods to elicit calibration from the predicted probability distributions.

In the third and final section, we will discuss the utility of calibration in downstream application settings. We will touch upon existing work which aim to bridge the gap between optimizing for calibration in UQ and deriving optimal utility when using the UQ models to drive decision-making. We make the observation that the aspect of UQ that actually matters will be dictated by the problem setting that uses the UQ model, and calibration may not be the most consequential metric in predicting the utility the UQ model will provide. We make the claim that the ultimate

utility of a predictive distribution or model is the performance it achieves in the downstream task it was designed for. Along this argument, we present several future research directions that specifically address using predictive distributions. One such application is in mixture of experts architectures, where we propose improving the routing mechanism for higher accuracy and/or efficiency in inference. We also propose re-visiting proper scoring rules as a method to elicit various utility from model predictions.

CHAPTER 2

Prior Work

2.1 Preliminaries and Background

We first lay out the notation, terminology, and class of models considered in this manuscript. Then we provide an overview of evaluation metrics in UQ and demonstrate how the pinball loss may be inadequate both as an evaluation metric and as an optimization objective.

2.1.1 Notation

Upper case letters X, Y denote random variables, lower case letters x, y, denote their values, and calligraphic upper case letters $\mathcal{X}, \mathcal{Y}$ denote sets of possible values. We use $x \in \mathcal{X}$ to denote the input feature vector and $y \in \mathcal{Y}$ to denote the corresponding target. Additionally, we consider the regression setting where $\mathcal{Y} \subset \mathbb{R}$ and $\mathcal{X} \subset \mathbb{R}^n$. We use $F_X, F_{Y|x}, F_Y$ to denote the true cumulative distribution of the subscript random variable. For any $x \in \mathcal{X}$, we assume there exists a true conditional distribution $F_{Y|x}$ over $\mathcal{Y}$, and we assume $Q_p(x)$ denotes the true p^{th} quantile of this distribution, i.e. $F_{\mathbf{Y}|x}(Q_p(x)) = p$. Any estimates of the true functions F, Q_p will be denoted with a hat, $\hat{F}, \hat{Q}_p$. We will specifically refer to any family of estimates for Q_p, with $p \in (0, 1)$, as a "quantile model", denoted $\hat{Q} : \mathcal{X} \times (0, 1) \to \mathcal{Y}$. Unless otherwise noted, we will always consider the *conditional* problem of estimating quantities in the target space $\mathcal{Y}$, conditioned on a value $x \in \mathcal{X}$.

2.1.2 Assessing the Quality of Predictive UQ

While various metrics have been proposed to assess the quality of UQ, there has been a great deal of recent focus on the notions of *calibration* and *sharpness* [Fasiolo et al., 2020, Cui et al., 2020, Zhao et al., 2020, Tran et al., 2020, Song et al., 2019, Kuleshov et al., 2018, Guo et al., 2017, Gneiting et al., 2007]. We introduce calibration here, but for a more thorough treatment, see Zhao et al. [2020]. Broadly speaking, calibration in the regression setting requires that the probability

of observing the target random variable below a predicted p^{th} quantile is equal to the *expected probability* p, for all $p \in (0, 1)$. We refer to the former quantity as the *observed probability* and denote it $p^{\text{obs}}(p)$, for an expected probability p, which we will write as p^{obs} when it is clear from context. Calibration requires $p^{\text{obs}}(p) = p$, $\forall p \in (0, 1)$. From this generic statement, we can describe different notions of calibration based on how p^{obs} is defined.

A model is **individually calibrated** if it outputs the true conditional quantiles, i.e. $\hat{Q}_p(x) = Q_p(x)$. In this case, we define the observed probability to be

$$p^{\text{obs}}_{indv}(p, x) := F_{\mathbf{Y}|x}(\hat{Q}_p(x)), \ \ \forall x \in \mathcal{X}, \ \ \forall p \in (0, 1). \tag{2.1}$$

In words, this requires that the probability of observing y below the quantile prediction is equal to p, *at each point* $x \in \mathcal{X}$*, individually*. If we can verify this property for all $x \in \mathcal{X}$, then by definition, we will know the quantile output is correct and precisely the true conditional quantile. However, individual calibration is typically unverifiable with finite datasets in the assumption-less case [Zhao et al., 2020].

A relaxed condition is **average calibration**. In this case, we define the observed probability to be

$$p^{\text{obs}}_{avg}(p) := \mathbb{E}_{x \sim F_{\mathbf{X}}}[F_{\mathbf{Y}|x}(\hat{Q}_p(x))], \ \ \forall p \in (0, 1), \tag{2.2}$$

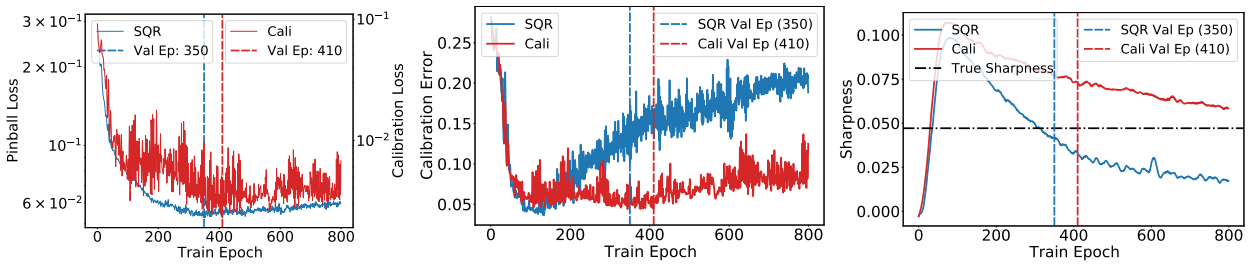
i.e. the probability of observing the target below the quantile prediction, *averaged over* $F_{\mathbf{X}}$, is equal to p. Average calibration is often referred to simply as "calibration" [Cui et al., 2020, Kuleshov et al., 2018]. Given a dataset $D = \{(x_i, y_i)\}_{i=1}^N$, we can estimate $p^{\text{obs}}_{avg}(p)$ with $\hat{p}^{\text{obs}}_{avg}(D, p) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}\{y_i \leq \hat{Q}_p(x_i)\}$.

Note that if our quantile estimate achieves average calibration then $\hat{p}^{\text{obs}}_{avg} \to p$ as $N \to \infty$, $\forall p \in (0, 1)$.

The degree of error in average calibration is commonly measured by *expected calibration error* Tran et al. [2020], Cui et al. [2020], Guo et al. [2017], $\text{ECE}(D, \hat{Q}) = \frac{1}{m} \sum_{j=1}^m \left| \hat{p}^{\text{obs}}_{avg}(D, p_j) - p_j \right|$, where $p_j \sim \text{Unif}(0, 1)$.

It may be possible to have an uninformative, yet average calibrated model. For example, quantile predictions that match the true *marginal* quantiles of $F_{\mathbf{Y}}$ will be average calibrated, but will hardly be useful since they do not depend on the input x. Therefore, the notion of **sharpness** is also considered, which quantifies the concentration of distributional predictions [Gneiting et al., 2007]. For example, for non-parametric predictions, the width of a centered 95% prediction interval is often used as a measure of sharpness. There generally exists a tradeoff between average calibration and sharpness [Gneiting et al., 2007, Murphy, 1973].

Recent works have suggested a notion of calibration stronger than average calibration, called



(a) Test Loss Curves (b) Test Calibration (c) Test Sharpness

Figure 2.1: **(a)** Test loss continues to decrease until the validated epoch. **(b-c)** At the validated epoch, *SQR* (optimizes *pinball loss*) is highly miscalibrated while sharper than the true sharpness level. *Cali* (optimizes proposed *calibration loss*) is better calibrated while less sharp than the true sharpness.

adversarial group calibration [Zhao et al., 2020]. This stems from the notion of **group calibration** [Hébert-Johnson et al., 2017, Kleinberg et al., 2016], which prescribes measurable subsets $\mathcal{S}_i \subset \mathcal{X}$ s.t. $P_{x\sim F_{\mathbf{X}}}(x \in \mathcal{S}_i) > 0, i = 1, \ldots, k$, and requires the predictions to be average calibrated within each subset. Adversarial group calibration then requires average calibration for *any subset of* $\mathcal{X}$ *with non-zero measure*. Denote $\mathbf{X}_{\mathcal{S}}$ as a random variable that is conditioned on being in the set $\mathcal{S}$. For **adversarial group calibration**, the observed probability is

$$p_{adv}^{\text{obs}}(p) := \mathbb{E}_{x\sim F_{\mathbf{X}_{\mathcal{S}}}}[F_{\mathbf{Y}|x}(\hat{Q}_p(x))], \quad \forall p \in (0,1), \quad \forall \mathcal{S} \subset \mathcal{X} \text{ s.t. } P_{x\sim F_{\mathbf{X}}}(x \in \mathcal{S}) > 0. \tag{2.3}$$

With a finite dataset, we can measure a proxy of adversarial group calibration by measuring the average calibration within all subsets of the dataset with sufficiently many points.

Intuitively, individual calibration inspects the discrepancy between p^{obs} and p for individual inputs $x \in \mathcal{X}$, adversarial group calibration relaxes this by inspecting any subset of $\mathcal{X}$ with non-zero measure, and average calibration relaxes this further by considering the full distribution of X.

One alternative family of evaluation metrics is **proper scoring rules** [Gneiting and Raftery, 2007]. Proper scoring rules are summary statistics of overall performance of a distributional prediction and consider both calibration and sharpness jointly [Gneiting et al., 2007]. For example, negative log-likelihood (NLL) is a proper scoring rule that is commonly used with density predictions [Detlefsen et al., 2019, Pearce et al., 2018a, Lakshminarayanan et al., 2017]. For quantile predictions, one proper score is the **check score**, *which is identical to the pinball loss.* Since proper scoring rules consider both calibration and sharpness together in a single value, they can serve as optimization objectives for UQ. For example, optimizing the pinball loss is the traditional method

in quantile regression [Koenker and Bassett Jr, 1978], and many recent quantile-based UQ methods focus on optimizing this objective [Rodrigues and Pereira, 2020, Tagasovska and Lopez-Paz, 2019, Cannon, 2018, Xu et al., 2017].

2.2 Quantile Methods for Calibrated Univariate Probabilistic Regression

This section is based on Chung et al. [2021].

Given that the definition of calibration is based on quantiles, we investigate the standard method of learning quantiles, which is the pinball loss. Specifically, we note that the balance between calibration and sharpness implied by the pinball loss is arbitrary and depends on the expressivity of the model class—and with highly expressive models, this balance can be heavily skewed towards sharpness. In their seminal work on probabilistic forecasts, Gneiting and Raftery [2007] contend that the goal of probabilistic forecasting is to "maximize the sharpness of the predictive distribution subject to calibration", i.e. calibration should be first achieved and then sharpness optimized. We show that common machine learning methods that use the pinball loss objective may in fact lead to an arbitrary and miscalibrated UQ.

Proposition 1. *Consider a finite dataset D, the pinball loss ρ_τ and a quantile model $f : \mathcal{X} \times (0,1) \to \mathcal{Y}$ that is average calibrated on D, i.e. ECE$(D, f) = 0$. Then there always exists another quantile model $g : \mathcal{X} \times (0,1) \to \mathcal{Y}$, such that, for any quantile level $\tau \in (0,1)$, g has lower pinball loss than f on D, i.e. $\sum_{i=1}^{N} \rho_\tau(y_i, g_\tau(x_i)) < \sum_{i=1}^{N} \rho_\tau(y_i, f_\tau(x_i))$, but worse average calibration than f, i.e. ECE$(D, g) >$ ECE(D, f).*

Proof: The proof is given in Appendix of Chung et al. [2021].

This proposition essentially states how the pinball loss can become detached from calibration, and we show its practical ramifications via a synthetic example in Figure 2.1 (experiment details in Appendix of Chung et al. [2021]).

We first note in Figure 2.1 (a) and (b) that even while the pinball loss decreases on the test set, test calibration worsens (while sharpness improves). Further, at the best validation epoch, optimizing the pinball loss converges to a solution that is sharper than the true noise level. Note that a UQ that is sharper than the true noise level will *never* be calibrated (meanwhile, a less sharp prediction *can still be calibrated*, e.g. the marginal distribution $F_{\mathbf{Y}}$). These pitfalls motivate our methods in Section 2.2.1.

2.2.1 Methods

We propose four methods that aim to produce an improved quantile model. The first is a model-agnostic procedure that relies on conditional density estimation (Section 2.2.1.1). To address settings where density estimation may be difficult, we then propose two loss functions to optimize with differentiable models: the combined calibration loss (Section 2.2.1.2), which directly optimizes calibration and sharpness, and the interval score (Section 2.2.1.3), which is a proper scoring rule for centered intervals. Finally, we propose a group batching method (Section 2.2.1.4) that can be applied to the batch optimization procedure for any loss function (e.g. combined calibration loss, interval score, and even pinball loss) to induce better convergence towards adversarial group calibration.

2.2.1.1 Utilizing Conditional Density Estimation for Model Agnostic QR

One drawback of many existing quantile-based UQ methods is that their training procedure requires differentiable models. In fact, most UQ methods require a specific class of models because of their modeling structure or their loss objective (e.g. Gaussian processes [Rasmussen, 2003], dropout [Gal and Ghahramani, 2016], latent variable models [Koller and Friedman, 2009], simultaneous pinball loss [Tagasovska and Lopez-Paz, 2019], and NLL-based losses [Lakshminarayanan et al., 2017]). This model restriction can be especially unfavorable in practical settings. A domain expert with an established point prediction model and compute infrastructure may want to add UQ without much additional overhead.

To address these issues, we can consider the following model-agnostic procedure. Instead of optimizing a designated loss function, we can consider splitting the given problem into two parts: estimate conditional quantiles directly from data, then regress onto these estimates. The benefit of this method is that, granted we can estimate the conditional quantiles accurately, we can use any regression model to regress onto these quantile estimates. Further, this regression task directly targets the goal of producing the true conditional quantiles (i.e. individual calibration). This procedure, which we refer to as *Model Agnostic QR* (MAQR), is outlined in Algorithm 1.

MAQR is based on the key assumption that nearby points in $\mathcal{X}$ will have similar conditional distributions, i.e. if $x_j \approx x_k$ then $F_{\mathbf{Y}|x_j} \approx F_{\mathbf{Y}|x_k}$. Given this smoothness assumption, we can group neighboring points to estimate the conditional density at each locality over $\mathcal{X}$, with locality determined by the hyperparameter d_N (Algorithm 2, line 2). We then construct an empirical CDF with the group of neighboring points, and conditional quantile estimates are produced with this empirical CDF. These estimates are collected into D (Algorithm 1, line 6), which is ultimately used as the training set for the quantile model $\hat{g}$.

In practice, we perform these steps with *residuals*, by first estimating a mean function $\hat{f}$ (Algorithm 1, line 1). This practical choice stems from existing works in conditional density estimation,

Algorithm 1 MAQR

1: **Input:** Train data $\{x_i, y_i\}_{i=1}^N$, trained regression model $\hat{f}(x)$
2: Calculate residuals $\epsilon_i = y_i - \hat{f}(x_i),\ i \in [N]$, and denote the residual dataset $R = \{x_i, \epsilon_i\}_{i=1}^N$
3: Initialize $D \leftarrow \varnothing$
4: **for** $i = 1$ **to** N **do**
5: $\quad D_i \leftarrow$ CondQuantilesEstimators(R, i) (Algorithm 2)
6: $\quad D \leftarrow D \cup D_i$
7: **end for**
8: Use D to fit a regression model $\hat{g}$
$\quad \hat{g} : (x, p) \mapsto \epsilon$
9: **Output:** $\hat{f} + \hat{g}$

Algorithm 2 CondQuantilesEstimators

1: **Input:** Dataset $\{x_i, \epsilon_i\}_{i=1}^N$, point index $k \in [N]$
2: $E_{k,d_N} \leftarrow \{\epsilon_i : \text{dist}(x_k, x_i) \leq d_N, i \in [N]\}$
3: Construct an empirical CDF with E_{k,d_N} to produce $\hat{F}_{\mathbf{E}|x_k} : \epsilon \mapsto p \in [0, 1]$
4: Initialize $D \leftarrow \varnothing$
5: **for** each ϵ_j in E_{k,d_N} **do**
6: $\quad \hat{p}_{k,j} \leftarrow \hat{F}_{\mathbf{E}|x_k}(\epsilon_j)$
7: $\quad D \leftarrow D \cup \{x_k, \hat{p}_{k,j}, \epsilon_j\}$
8: **end for**
9: **Output:** D

which suggests that having 0 conditional mean in the data provides benefits in terms of lower asymptotic mean squared error in the conditional density predictions [Hyndman et al., 1996]. Further, this demonstrates how MAQR can be readily applied in the application setting where an accurate point prediction model often already exist.

Algorithm 1 is a specific implementation of a more general model-agnostic algorithm, in which we directly estimate conditional quantiles from the data with tools from conditional density estimation. We note that using KDEs for conditional density estimation is a well studied problem with theoretical guarantees [Holmes et al., 2012, Hyndman et al., 1996, Stute et al., 1986]. In the case the distance in $\mathcal{X}$ is measured using a uniform kernel with mild assumptions on the bandwidth, Algorithm 1 falls under the guarantees stated by Stute et al. [1986].

Theorem 1 [Stute et al., 1986]. *Assume $\mathcal{Y} \subset \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^n$, dist$(x_i, x_j) := |x_i - x_j|_\infty$, and that $\hat{F}_{\mathbf{E}|x}$ is constructed using the procedure given in line 5 of Algorithm 1 (i.e. $x_i = x$). Further assume that, as $N \to \infty$, $d_N \to 0$ and that $\sum_{N \geq 1} \exp(-\rho N d_N^n) < \infty, \forall \rho > 0$. Then, as $N \to \infty$, for almost all $x \in \mathcal{X}$, $\sup_\epsilon[\hat{F}_{\mathbf{E}|x}(\epsilon) - F_{\mathbf{E}|x}(\epsilon)] \to 0$ with probability 1.*

This theorem states that in the limit of data, for almost all $x \in \mathcal{X}$, the CDF estimate $\hat{F}_{\mathbf{E}|x}$ will converge uniformly to the true CDF $F_{\mathbf{E}|x}$ with probability 1. The dataset, D, will therefore be populated with good estimates of the conditional quantile and quantile level pair for x.

2.2.1.2 Explicitly balancing calibration and sharpness with the combined calibration loss

While MAQR can produce strong results, its performance can suffer in high-dimensional settings, where nonparametric conditional density estimation methods falter. Neural networks (NNs) have shown good performance in high dimensional settings, given their high capacity to approximate complex functions and recent advances in fast gradient-based optimization. We therefore propose a loss-based approach to estimating conditional quantiles for NNs and other differentiable models.

Drawing motivation from the *arbitrary* balance between calibration and sharpness that pinball loss *implicitly* provides, we propose objectives separately for calibration and sharpness, Then, we combine the two objectives into a single loss function that provides an explicit balance between calibration and sharpness that can be chosen by the end user.

We first consider calibration of a quantile prediction, $\hat{Q}_p \in \mathcal{Y}$ for quantile level $p \in (0, 1)$. Here, we omit conditioning on x for clarity. For this prediction to be average calibrated, exactly a p proportion of the true density should lie below $\hat{Q}_p$, i.e. $p_{avg}^{\text{obs}} = P(Y \leq \hat{Q}_p) = p$. While calibration (e.g. $|p_{avg}^{\text{obs}} - p|$) is a non-differentiable objective, by inducing a truncated distribution based on the current level of calibration, we can construct the following calibration objective, which is minimized if and only if the prediction is average calibrated:

$$\begin{aligned}\mathcal{C}(\hat{Q}_p, p) = \mathbb{I}\{\hat{p}_p < p\} * \mathbb{E}[Y - \hat{Q}_p | Y > \hat{Q}_p] * P(Y > \hat{Q}_p) \\ + \mathbb{I}\{\hat{p}_p > p\} * \mathbb{E}[\hat{Q}_p - Y | \hat{Q}_p > Y] * P(\hat{Q}_p > Y), \text{where } \hat{p}_p = P(Y \leq \hat{Q}_p).\end{aligned} \tag{2.4}$$

The empirical calibration objective, $\mathcal{C}(D, \hat{Q}_p, p)$, is then defined as follows:

$$\begin{aligned}\mathcal{C}(D, \hat{Q}, p) = \mathbb{I}\{\hat{p}_{avg}^{\text{obs}} < p\} * \frac{1}{N} \sum_{i=1}^{N} \left[(y_i - \hat{Q}_p(x_i)) \mathbb{I}\{y_i > \hat{Q}_p(x_i)\} \right] \\ + \mathbb{I}\{\hat{p}_{avg}^{\text{obs}} > p\} * \frac{1}{N} \sum_{i=1}^{N} \left[(\hat{Q}_p(x_i) - y_i) \mathbb{I}\{\hat{Q}_p(x_i) > y_i\} \right].\end{aligned} \tag{2.5}$$

Note 1: *Intuition of the calibration objective.* For any given p, consider the case when the quantile estimate $\hat{Q}_p$ is below the true p^{th} quantile $\mathbb{Q}_p$. Since $\hat{Q}_p < \mathbb{Q}_p \implies \hat{p}_p < p$, this implies that too much data density lies above $\hat{Q}_p$. In this case, $\mathcal{C}(\hat{Q}_p, p)$ reduces to $\mathbb{E}[Y - \hat{Q}_p | Y > \hat{Q}_p] * P(Y > \hat{Q}_p)$. $\hat{Q}_p$ is pulled higher with the expectation of the truncated distribution that places $\hat{Q}_p$ at the lower bound of the support. In the opposite case, when $\hat{Q}_p > \mathbb{Q}_p$, $\hat{Q}_p$ is pulled lower by the same logic.

Note 2: *Is the proposed calibration objective a proper scoring rule?* Strictly speaking, the calibration objective is a non-decomposable function, hence deviates from the standard convention

of proper scoring rules Gneiting and Raftery [2007], which can be "decomposed" into scores for individual examples (x_i, y_i). This simply arises from the fact that measuring average calibration (i.e. $\hat{p}^{\text{obs}}_{avg}$) is non-decomposable. Proper scoring rules are defined such that an optimum of the *expected score* (or *risk*, if we consider the score as a *loss function*) occurs at the true distribution quantity. While an example level *loss* or *score* does not exist due to non-decomposability, we can still show the (expectation-level) score (i.e. $\mathcal{C}(\hat{Q}_p, p)$) is minimized by the true distribution and hence enjoys the optimum property of proper scoring rules.

Proposition 2. *For any quantile level $p \in (0, 1)$, the true quantile function $\mathbb{Q}_p$ minimizes the calibration objective, $\mathcal{C}(\hat{\mathbb{Q}}_p, p)$. Further, on a finite dataset D, the empirical calibration objective, $\mathcal{C}(D, \hat{\mathbb{Q}}_p, p)$, is minimized by an average calibrated solution on D, i.e. when $\hat{p}^{obs}_{avg}(D, p) = p$.*

Proof: The proof is given in Appendix of Chung et al. [2021].

***Note 3:** Non-zero gradients for miscalibrated predictions $\hat{Q}_p$.* We can further show that for a miscalibrated quantile prediction, the gradients of $\mathcal{C}$ are always non-zero. When $\hat{p}_p < p$, $\partial\mathcal{C}(\hat{Q}_p, p)/\partial\hat{Q}_p = -P(Y > \hat{Q}_p) < 0$. Thus increasing $\hat{Q}_p$ decreases the objective $\mathcal{C}$. Similarly, when $\hat{p}_p > p$, $\partial\mathcal{C}(\hat{Q}_p, p)/\partial\hat{Q}_p = P(Y < \hat{Q}_p) > 0$, and an analogous argument follows (proof in Appendix of Chung et al. [2021]).

As discussed in Section 2.1.2, average calibration by itself is not a sufficient condition for meaningful UQ, hence we also desire *sharp* quantile models, with more-concentrated (less dispersed) distributions. We can induce this property in quantile predictions by predicting the $(1-p)^{\text{th}}$ quantile $\hat{Q}_{1-p}(x_i)$ alongside each prediction $\hat{Q}_p(x_i)$ and penalizing the width between the quantile predictions:

$$\mathcal{P}(\hat{Q}_p, p) = \mathbb{E}\left[\left|\hat{Q}_p - \hat{Q}_{1-p}\right|\right]. \tag{2.6}$$

The empirical sharpness objective, $\mathcal{P}(D, \hat{Q}_p, p)$, is then defined as follows:

$$\mathcal{P}(D, \hat{Q}, p) = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} \hat{Q}_{1-p}(x_i) - \hat{Q}_p(x_i) \;\; (p \leq 0.5) \\ \hat{Q}_p(x_i) - \hat{Q}_{1-p}(x_i) \;\; (p > 0.5). \end{cases} \tag{2.7}$$

It is important to note that the true underlying distribution will not have 0 sharpness if there is significant noise, and sharpness should be optimized subject to calibration. Therefore, we should only penalize sharpness when the data suggests our quantiles are too dispersed, i.e. when $\left|p^{\text{obs}}_{avg}(p) - p^{\text{obs}}_{avg}(1-p)\right|$, the *observed coverage* between the pair of quantiles $\hat{Q}_p(x_i)$ and $\hat{Q}_{1-p}(x_i)$, is greater than $|2p-1|$, the *expected coverage*.

Combining the calibration and sharpness terms, we have the **combined calibration loss**

$$\mathcal{L}(D, \hat{Q}_p, p) = (1 - \lambda)\mathcal{C}(D, \hat{Q}_p, p) + \lambda\mathcal{P}(D, \hat{Q}_p, p). \tag{2.8}$$

The hyperparameter $\lambda \in [0, 1]$ sets the explicit balance between calibration and sharpness. Note that setting $\lambda = 0$ may not always be desirable, since optimizing $\mathcal{C}(D, \hat{Q}_p, p)$ alone may converge to quantiles of the marginal distribution, $F_{\mathbf{Y}}$. Further, in certain downstream applications that utilize UQ, a sharper prediction, even at the cost of worse calibration, may result in higher utility, and λ can be tuned according to the utility function of the application. In our experiments, we tune λ by cross-validating with adversarial group calibration as it is the strictest notion of calibration that can be estimated with a finite dataset. Since we learn a quantile model that outputs the conditional quantile estimates for all probabilities, our training objective is $\mathbb{E}_{p\sim\text{Unif}(0,1)}\mathcal{L}(D, \hat{Q}_p, p)$.

2.2.1.3 Encouraging calibration of centered intervals with the interval score

The combined calibration loss (Eq. 2.8) optimizes average calibration, which targets observed probabilities below a quantile. In many applications, however, we may desire a centered prediction interval (PI) which requires a pair of quantile predictions. A centered 95% PI, for example, is a pair of quantile predictions at quantile levels 0.025 and 0.975. Hence, for the average calibration of the p^{th} *centered interval*, we want $\left[\hat{p}^{\text{obs}}_{avg}(0.5 + \frac{p}{2}) - \hat{p}^{\text{obs}}_{avg}(0.5 - \frac{p}{2})\right]$ (the PI's observed probability, a.k.a. prediction interval coverage probability (PICP) [Tagasovska and Lopez-Paz, 2019, Kabir et al., 2018, Pearce et al., 2018b]) to be equal to the expected probability p. While we can modify the objective in Eq. 2.8 to adhere to this altered goal, here we propose simultaneously optimizing the **interval score** (or Winkler score) [Gneiting and Raftery, 2007, Winkler, 1972] for all expected probabilities $p \in (0, 1)$, and bring to light a proper scoring rule that has largely been neglected for the purpose of *learning quantiles*. While some previous works utilize the interval score to *evaluate* interval predictions [Bracher et al., 2021, Bowman et al., 2020, Askanazi et al., 2018, Maciejowska et al., 2016], to the best of our knowledge, no previous work has focused on simultaneously optimizing it and shown a thorough experimental evaluation.

For a point (x, y), if we denote a $(1 - \alpha)$ centered PI as $\hat{l}, \hat{u}$, i.e. $\hat{l} = \hat{Q}_{\frac{\alpha}{2}}(x)$ and $\hat{u} = \hat{Q}_{1-\frac{\alpha}{2}}(x)$, the interval score is defined as $S_\alpha(\hat{l}, \hat{u}; y) = (\hat{u} - \hat{l}) + \frac{2}{\alpha}(\hat{l} - y)\mathbb{I}\{y < \hat{l}\} + \frac{2}{\alpha}(y - \hat{u})\mathbb{I}\{y > \hat{u}\}$. We show in Appendix of Chung et al. [2021] that the minimum of the expectation of the interval score is attained at the true conditional quantiles, $\hat{l} = Q_{\frac{\alpha}{2}}(\cdot)$, $\hat{u} = Q_{1-\frac{\alpha}{2}}(\cdot)$. We train our quantile model for all centered intervals (and hence all quantile levels) simultaneously by setting our loss as $\mathbb{E}_{\alpha\sim\text{Unif}(0,1)}S_\alpha$.

2.2.1.4 Inducing adversarial group calibration with group batching

The calibration loss (Section 2.2.1.2) and the interval score (Section 2.2.1.3) optimize for the *average* calibration of quantiles and centered intervals, respectively. To get closer to individual calibration, one condition we can additionally require is *adversarial group calibration*. Since adversarial group calibration requires average calibration over any subset of non-zero measure over the domain, this is not fully observable with finite datasets D for all subset sizes. However, for any subset in D with enough datapoints, we can still estimate average calibration over the subset. Hence, we can apply our optimization objectives onto appropriately large subsets to induce adversarial group calibration.

In practice, this involves constructing subsets within the domain and taking gradient steps based on the loss over each subset. In naive implementations of stochastic gradient descent, a random batch is drawn *uniformly* from the training dataset D, and a gradient step is taken according to the loss over this batch. This is also the case in *SQR* [Tagasovska and Lopez-Paz, 2019]. The uniform draw of the batch will tend to preserve $F_{\mathbf{X}}$ (the marginal distribution of $\mathbf{X}$), hence optimizing average calibration over this batch will only induce average calibration of the model.

Instead, deliberately grouping the datapoints based on input features, and then batching and taking gradient steps based on these batches, induces better adversarial group calibration. We find in our experiments that adversarial group calibration improves significantly with simple implementations of group batching.

To summarize, the main idea we introduce here with group batching is that, only taking uniform batches from the training set (thus only drawing batches which preserve $F_{\mathbf{X}}$) can be detrimental when optimizing for calibration. Thus, additionally drawing batches based on deliberate groupings within the training set (thus, batches which do not preserve $F_{\mathbf{X}}$) can help to induce a stronger notion of calibration (i.e. adversarial group calibration) in the model than average calibration. This concept is quite general and allows for variations in implementations when constructing the groups.

2.3 Calibration for Multi-dimensional Probabilistic Regression

This section is based on Chung et al. [2024].

We begin our discussion from the observation that the most widely studied notions of calibration in regression are usually confined to the setting where the targets are single dimensional [Gneiting et al., 2007, Pearce et al., 2018b, Kuleshov et al., 2018, Song et al., 2019, Cui et al., 2020,

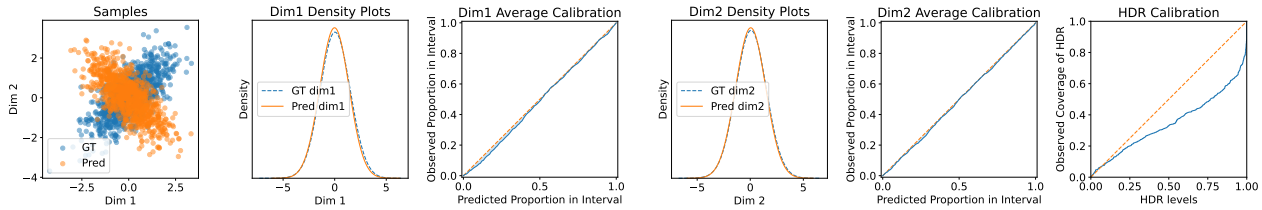


Figure 2.2: Pitfall of assessing the calibration of each dimension independently for multi-dimensional distributional predictions. **(From Left to Right)** The predictive distribution (labeled *Pred*) exhibits the opposite correlation in the output dimensions compared to the ground truth (labeled *GT*), but each of the marginal distributions are accurate. Assessing calibration of each dimension separately suggests a well-calibrated predictive distribution. Highest density regions (HDRs) are able to account for the dependence in the dimensions, and assessing HDR calibration reveals the miscalibration of the joint distribution.

Zhao et al., 2020, Sahoo et al., 2021b, Kuleshov and Deshpande, 2022]. While multi-dimensional regression models are widely used in machine learning, especially in applications such as model-based control [Chua et al., 2018, Malik et al., 2019, Yu et al., 2020, Kidambi et al., 2020] or modeling in the physical sciences [Sexton et al., 2012, Duraisamy et al., 2019, Abbate et al., 2021, Char et al., 2023a], we find that methods which account for the joint multi-dimensional distribution in *assessing* calibration and *recalibrating* the prediction is generally lacking. In lieu of more sophisticated methods, calibration is often considered for each output dimension independently.

However, failing to account for interplay among the output dimensions may be problematic when dependence does exist. In this case, the collection of marginals is not sufficient to provide an accurate assessment of the prediction quality (see Figure 2.2 for an example).

In this work, we address the problem of calibration in multi-dimensional regression by first formalizing a notion of calibration which *can* account for dependence among the output dimensions and further proposing a recalibration algorithm for the joint predictive distribution. We summarize our main contributions as follows:

- By leveraging existing ideas in highest density regions (HDR), we propose the notion of *HDR calibration*, which accounts for dependence in the output dimensions in defining and evaluating calibration for multi-dimensional distributional predictions.
- We develop a recalibration algorithm for multi-dimensions which produces HDR calibrated predictive distributions via a sampling procedure.
- We provide extensive demonstrations of the merits of the notion of HDR calibration and the efficacy of the recalibration algorithm on a suite of benchmark datasets in multi-dimensional

regression, and two real-world datasets: a dynamics modeling task in nuclear fusion, and a downstream decision-making application in forecasting customer demand.

We proceed by first describing the problem setting and relevant concepts to motivate the definition of HDR calibration in Section 2.1. Based on this notion of calibration, we present our proposed HDR recalibration algorithm in Section 2.3.5.

2.3.1 Setting and Notation

We consider the regression setting with an input feature space $\mathcal{X} \subseteq \mathbb{R}^n$ and a target space $\mathcal{Y} \subseteq \mathbb{R}^D$. We use x^d, X^d, y^d, and Y^d to denote the d^{th} dimension of input and target vectors. f and F denote the true probability density function (PDF) and cumulative distribution function (CDF), and when it exists, we denote the true quantile function with F^{-1}. Estimates of these functions are denoted with $\hat{f}$, $\hat{F}$ and $\hat{F}^{-1}$. We use subscripts to indicate the corresponding random variable of the PDFs and CDFs (e.g. f_X and F_X are the marginal PDF and CDF of X, and $f_{Y|X}$ and $F_{Y|X}$ are the PDF and CDF of Y conditioned on X). When conditioning on a specific value $X = x$, we denote the conditional distribution functions as $f_{Y|x}$ and $F_{Y|x}$. Lastly, we assume that new target samples can be drawn from the distribution estimate, and we denote the random variable corresponding to these new target samples as $\hat{Y}$. In particular, this can be done by sampling $X \sim f_X$ from the dataset and subsequently sampling $\hat{Y}|X \sim \hat{f}_{Y|X}$. Importantly, note that the distribution of $\hat{Y}$ is still tied to the distribution of X.

2.3.2 Calibration in Univariate Regression

Before discussing the multi-dimensional setting, we first provide a brief review of notions of calibration in the univariate setting. A widely accepted notion of calibration in univariate regression is *probabilistic calibration* [Gneiting et al., 2007]. A predictive distribution $\hat{F}_{Y|X}$ is probabilistically calibrated if

$$P(Y \leq \hat{F}^{-1}_{Y|X}(p)) = p, \forall p \in (0, 1). \tag{2.9}$$

This notion is also referred to as simply *calibration* [Kuleshov et al., 2018], *quantile calibration* [Song et al., 2019], or *average calibration* [Zhao et al., 2020, Chung et al., 2021] since it focuses on the average validity of the predictive quantile function $\hat{F}^{-1}_{Y|X}$. We henceforth refer to this notion as *average calibration*. Here, we note that the true distribution $F_{Y|X}$ trivially satisfies Eq. 2.9 since $F_{Y|X}(Y) \sim \mathcal{U}(0, 1)$ by the probability integral transform and $P(\hat{F}_{Y|X}(Y) \leq p) = p$ is the CDF of $\mathcal{U}(0, 1)$.

From this general definition, subsequent works have derived various notions of calibration, usually by placing different conditions in assessing the empirical probability (LHS of Eq. 2.9).

For example, *distribution calibration* assesses average calibration conditioned on the predictive distribution; *individual calibration* [Zhao et al., 2020] requires average calibration conditioned on each input point, $x \in \mathcal{X}$; and *group calibration* [Kleinberg et al., 2016, Hébert-Johnson et al., 2017] requires average calibration conditioned on specific subsets of the input space with non-zero measure.

In all of the aforementioned notions, Y is assumed to be univariate (i.e $\mathcal{Y} \subseteq \mathbb{R}$), and predictive conditional quantiles $\hat{F}^{-1}_{Y|X} : \mathcal{X} \times (0, 1) \to \mathcal{Y}$ are utilized to measure the discrepancy between predicted and empirical probabilities (RHS and LHS of Eq. 2.9).

2.3.3 The Multi-dimensional Setting

While a naive application of the notions of univariate calibration to multi-dimensional distribution functions may seem plausible, in the multivariate setting, the quantile function is not well-defined [Belloni and Winkler, 2009], and further, $F_Y(Y)$ for $Y \in \mathbb{R}^D, D > 1$ is no longer uniformly distributed [Barbe et al., 1996, Genest and Rivest, 2001]. To circumvent these issues, prior works have suggested utilizing projections of the target variable Y in order to *define* and *assess* calibration of multi-dimensional distributional predictions. We formalize such methods as follows.

Consider a mapping $g : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{Z} \subseteq \mathbb{R}$.

Furthermore, we let Z and $\hat{Z}$ be the r.v.s over the projection outputs when using target labels Y and $\hat{Y}$, respectively. Concretely, $Z \coloneqq g(X, Y)$ and $\hat{Z} \coloneqq g(X, \hat{Y})$. Since sampling from the predicted distribution is cheap, we can estimate the CDF $F_{Z|X}$ using the empirical distribution of $\hat{Z}|X$. We refer to this empirical CDF as $\hat{F}_{Z|X}$.

Then, following the definition of average calibration (Eq. 2.9), we can define calibration in the projected space as satisfying, $\forall p \in (0, 1)$,

$$P(Z \leq \hat{F}^{-1}_{Z|X}(p)) = p \tag{2.10}$$

$$\text{or equivalently, } P(\hat{F}_{Z|X}(Z) \leq p) = p. \tag{2.11}$$

We can easily show that the optimal prediction $\hat{F}_{Y|X} = F_{Y|X}$ satisfies this definition of calibration in the projected space.

Proposition 1 *The optimal distributional prediction, i.e. $\hat{F}_{Y|X} = F_{Y|X}$, satisfies calibration in the projected space, Eq. 2.11. (proof in Appendix of Chung et al. [2024])*

Several prior works have proposed specific versions of Eq. 2.11 with specific projection functions. Ziegel and Gneiting [2014] introduced *copula calibration* by utilizing the predictive CDF

as the projection function (i.e. $g(X, \cdot) = \hat{F}_{Y|X}$). In this specific case, the distribution of the projections is called the Kendall distribution [Nelsen et al., 2003].

One can also utilize the predictive PDF for the projection function such that $g(X, \cdot) = \hat{f}_{Y|X}$, in which case Eq. 2.11 bears intrinsic relationships to existing concepts of highest predictive density (HPD) values [Harrison et al., 2015, Dalmasso et al., 2020, Zhao et al., 2021a] and highest density regions (HDR) [Hyndman, 1996].

While there are several candidates for projection functions, in this work, we choose to focus on using the predictive PDF. In particular, we leverage its connections with HPD and HDR to formalize a notion of calibration in multi-dimensions (Defn. 1) and propose a recalibration procedure that achieves this notion of calibration (Section 2.3.5). Hence, in the rest of this work, we always assume $Z \coloneqq \hat{f}_{Y|X}(Y)$ and $\hat{Z} \coloneqq \hat{f}_{Y|X}(\hat{Y})$.

For any given (x, y), $\text{HPD}_x(y)$ is a measure of how plausible y is w.r.t $\hat{f}_{Y|x}$ and is defined as

$$\text{HPD}_x(y) = \int_{y' : \hat{f}_{Y|x}(y') \geq \hat{f}_{Y|x}(y)} \hat{f}_{Y|x}(y') dy'. \tag{2.12}$$

In words, $\text{HPD}_x(y)$ is the predicted probability of observing $\hat{Y}$ that is more likely than y, where the likelihood is determined by $\hat{f}_{Y|x}$. Considering the definition of Z and $\hat{Z}$, we see that

$$\begin{aligned} &\text{HPD}_x(y) && (2.13)\\ &= P(\hat{f}_{Y|x}(\hat{Y}) \geq \hat{f}_{Y|x}(y) \mid X = x) && (2.14)\\ &= 1 - \hat{F}_{Z|x}(\hat{f}_{Y|x}(y)). && (2.15) \end{aligned}$$

Further, HPD values, which are *probabilities*, have a direct relationship to HDRs, which are *prediction sets*.

For clarity, we provide a definition of HDR below using our notation, and we refer the reader to Appendix of Chung et al. [2024] for the original notation by Hyndman [1996]. For a fixed x and constant $\lambda \in \mathbb{R}$, we define the λ-density region as $\text{DR}_x(\lambda) \coloneqq \{y : \hat{f}_{Y|x}(y) \geq \lambda\}$.

Then for a given coverage level p, the p-HDR is the smallest density region with probability greater than or equal to p. Concretely,

$$\begin{gathered} \text{HDR}_x(p) \coloneqq \text{DR}_x(\lambda^*) \\ \text{where} \quad \lambda^* = \sup\{\lambda : P(\hat{Y} \in \text{DR}_x(\lambda) | X = x) \geq p\}. \end{gathered}$$

By their definitions, the following equivalence holds:

$$\text{HPD}_x(y) \leq p \iff y \in \text{HDR}_x(p) \tag{2.16}$$

We note that calibration is generally defined in terms of prediction sets of a distribution, and drawing on the intrinsic relationships between HDR, HPD and Eq. 2.15, we formalize Eq. 2.11 with the notion of HDR calibration:

Definition 1 *A predictive PDF $\hat{f}_{Y|X}$ is HDR calibrated if,* $\forall p \in (0, 1)$,

$$P(Y \in \textit{HDR}_X(p)) = p \tag{2.17}$$

$$\textit{or equivalently, } P(\textit{HPD}_X(Y) \leq p) = p. \tag{2.18}$$

Proposition 2 *HDR calibration holds if and only if Equation 2.11 holds. (proof in Appendix of Chung et al. [2024])*

Similar to average calibration (Eq. 2.9), which requires Y to be contained in the p-prediction set (defined by the p^{th} quantile) with probability p, HDR calibration requires the p-HDR to contain Y with probability p.

Utilizing projections allows one to *define* notions of calibration in the multi-dimensional setting which can account for dependence in the output dimension, granted that the projection function models the dependence. Further, based on the definitions, one can *assess* calibration (or miscalibration) via the discrepancy between the predicted and empirical probabilities. Following the commonly used notion of expected calibration error (ECE) [Guo et al., 2017, Cui et al., 2020, Tran et al., 2020, Chung et al., 2021] we can measure the (L1-)ECE w.r.t the general notion of calibration defined in Eq. 2.11 as

$$\mathbb{E}_{p\sim\mathcal{U}(0,1)} \left| P(\hat{F}_{Z|X}(Z) \leq p) - p \right|. \tag{2.19}$$

Not only do these metrics allow one to evaluate the quality of uncertainty for multi-dimensional predictions, but they can also be used to improve a model's predictive distribution via recalibration.

2.3.4 Recalibration

Probabilistic models are usually trained by optimizing a loss function which may not be necessarily aligned with calibration. This can often lead to models being miscalibrated at the end of the training [Guo et al., 2017, Kuleshov et al., 2018, Chung et al., 2021].

A *post-hoc recalibration* step can be applied on top of the trained model to adjust for its level of miscalibration observed on a held-out calibration or validation dataset.

Post-hoc recalibration is well-studied in classification, and there are several methods which have proven to be effective in producing well-calibrated (discrete) class probabilities [Platt et al., 1999, Zadrozny and Elkan, 2001, 2002, Guo et al., 2017, Gupta and Ramdas, 2021].

This problem is not as widely studied in regression, however, and to the best of our knowledge, the most popular method is that of Kuleshov et al. [2018], which learns an isotonic mapping between expected and observed quantile levels. Crucially, this method readily applies only to the case when the targets Y are univariate, and we henceforth refer to this algorithm as "single dimensional (SD) recalibration". In Section 2.3.5, we propose a recalibration procedure for the multivariate setting.

While also proposed for the single dimensional setting, it is worth mentioning that Izbicki et al. [2022] proposes a conformal prediction method which bears relevance as their method utilizes HPD values as the conformity score. However, there are key differences: while they are focused on producing prediction *sets* for a fixed coverage level (as is the goal of conformal prediction), we are focused on expressing the full predictive *distribution*. Crucially, since Izbicki et al. [2022] does not consider multi-dimensional target spaces, their method does not account for dependence in the target dimensions, and the algorithm relies on constructing a finite grid of the target space, which is ill-suited for higher dimensions. As we will discuss in Section 2.3.5, our recalibration procedure explicitly addresses dependence in the target dimensions and is more scalable as it focuses on *sampling* from a predictive distribution in the multi-dimensional space.

2.3.5 Method

In this section, we describe our proposed recalibration procedure which aims to achieve HDR calibration (Defn. 1). We describe the procedure in two parts. Section 2.3.5.1 details the recalibration algorithm that aims to optimize for Eq. 2.11, which is equivalent to HDR calibration by Proposition 2. Afterwards, Section 2.3.5.2 describes a pre-conditioning step that can modify the predictive PDF to account for dependence in the output dimensions when applying the recalibration algorithm.

2.3.5.1 HDR Recalibration Algorithm

The proposed recalibration algorithm is comparable to that of Kuleshov et al. [2018] for univariate settings, but with key differences – the recalibration occurs in the projected space $\mathcal{Z}$, and the recalibration output must be translated back into the target space $\mathcal{Y}$.

First, we estimate a recalibration mapping in the projected space by using observations of the random variable $\hat{F}_{Z|X}(Z)$ with a calibration dataset $\{(x_i, y_i)\}_{i=1}^N$, i.e. the observed values are $\{\hat{F}_{Z|x_i}(z_i)\}_{i=1}^N$ where $z_i = \hat{f}_{Y|x_i}(y_i)$. To elaborate more on this procedure, note that z_i is a

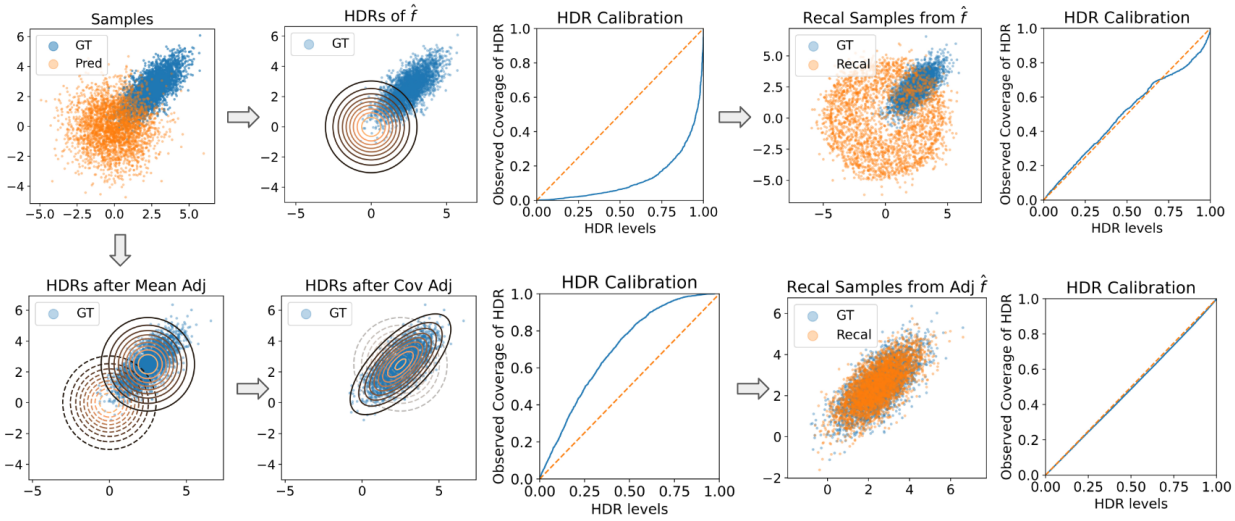


Figure 2.3: Demonstration of HDR recalibration on a marginal distributional prediction. **(Top Left)** The initial prediction (labeled *Pred*) displays bias in the mean prediction and fails to model the correlation in the ground truth distribution (labeled *GT*). **(Top Row)** Without the PDF adjustment step, we observe that observations (GT points) fall more often in the higher level HDRs (level sets defined by darker boundaries) than lower level HDRs (level sets bounded by lighter colors). HDR recalibration re-samples from each HDR according to the observed frequencies (i.e. the learned recalibration mapping), hence when producing recalibrated samples, the higher level HDRs (i.e. outer level sets of $\hat{f}$) are over-sampled and the lower level HDRs (inner level sets of $\hat{f}$) are under-sampled. The resulting recalibrated samples are HDR calibrated (right-most plot), but we can visually assess that the samples are suboptimal and in particular, fail to model the correlation in the dimensions. **(Bottom Row)** Before the recalibration procedure, we can estimate the bias in the mean on the calibration dataset and correlation in the dimensions with the correlation matrix of the mean prediction error. After applying these two adjustments, HDR calibration reveals that each p-HDR contains more than p proportion of the observations (which also indicates that the level sets are too wide). Hence, HDR recalibration proportionately under-samples from each HDR, which results in well-calibrated samples that also reflect the correlation in the output dimensions.

scalar value produced by evaluating the PDF $\hat{f}_{Y|x_i}$ at y_i, where $\hat{f}_{Y|x_i}$ is the PDF of the predictive distribution. $\hat{F}_{Z|x_i}(z_i)$ is also a scalar value produced by evaluating the CDF $\hat{F}_{Z|x_i}$ at z_i, however, $\hat{F}_{Z|x_i}$ is an empirical CDF over the projected space that is estimated by producing samples from the predictive distribution $\hat{f}_{Y|x_i}$. Again, we note that sampling from the predictive distribution is cheap, thus estimating this empirical CDF is also cheap.

Afterwards, we learn the monotonic mapping $R : [0, 1] \to [0, 1]$ where $R(p) \coloneqq P(\hat{F}_{Z|X}[Z] \leq p)$. R is then applied to the predictive distribution at each x, $\hat{F}_{Z|x}$, to produce the recalibrated predictive distribution $R \circ \hat{F}_{Z|x}$.

Algorithm 3 HDR Recalibration: Training

1: **Input**: Calibration dataset $\{(x_i, y_i)\}_{i=1}^N$, predictive PDF $\hat{f}_{Y|X}$.
2: $\hat{f}_{Y|X} \leftarrow \text{ADJUST}(\hat{f}_{Y|X})$.
3: Construct the dataset $\mathcal{C} = \{\hat{F}_{Z|x_i}(z_i)\}_{i=1}^N$, where $z_i = \hat{f}_{Y|x_i}(y_i)$.
4: Sort values in $\mathcal{C}$ to construct $\{c_{(i)}\}_{i=1}^N$, construct the recalibration dataset $\mathcal{C}' = \{i/N, c_{(i)}\}_{i=1}^N$.
5: Learn the recalibration mapping R on $\mathcal{C}'$.
6: **Output**: Recalibration mapping R.

Algorithm 4 HDR Recalibration: Sampling

1: **Input**: Test point x, predictive PDF $\hat{f}_{Y|X}$, recalibration mapping R, number of samples M.
2: $\hat{f}_{Y|X} \leftarrow \text{ADJUST}(\hat{f}_{Y|X})$.
3: Construct $\mathcal{D} = \{(\hat{y}_j, \hat{z}_j)\}_{j=1}^M$ by producing M samples $\hat{y}_j \sim \hat{f}_{Y|x}$ and setting $\hat{z}_j = \hat{f}_{Y|x}(\hat{y}_j)$.
4: Re-sample from $\mathcal{D}$ to construct $\mathcal{D}' = \{(\hat{y}_k, \hat{z}_k)\}_{k=1}^M$ s.t. $\{\hat{z}_k\}_{k=1}^M$ approximately follows $R \circ \hat{F}_{Z|x}$.
5: **Output**: Recalibrated samples at x, $\{\hat{y}_k\}_{k=1}^M$.

Proposition 3 *Consider $R \circ \hat{F}_{Z|X}$ for an invertible mapping R. Then $R \circ \hat{F}_{Z|X}$ satisfies Eq. 2.11, i.e.*

$$P(R \circ \hat{F}_{Z|X}(Z) \leq p) = p \quad \forall p \in (0, 1).$$

(proof in Appendix of Chung et al. [2024])

One can therefore use such a recalibration map, R, to draw new, calibrated samples in $\mathcal{Z}$ space. However, it remains unclear how to relate these samples back to their counterparts in $\mathcal{Y}$ space. To address this issue, we present a sampling algorithm that operates over samples of r.v. $\hat{Y}$. The key idea is to re-sample from the set of samples generated from $\hat{f}_{Y|X}$ according to what the distribution should look like in $\mathcal{Z}$ space. In particular, for any fixed x, we can draw many samples from the predictive PDF, $\{\hat{y}_j\}_{j=1}^M \sim \hat{f}_{Y|x}$, then apply the projection $\hat{f}_{Y|x}(\cdot)$ to produce the dataset of tuples $\mathcal{D} = \{(\hat{y}_j, \hat{z}_j)\}$, where $\hat{z}_j = \hat{f}_{Y|x}(\hat{y}_j)$, and note that by definition, $\hat{z}_j \sim \hat{f}_{Z|x}, \hat{F}_{Z|x}$. We then re-sample from $\mathcal{D}$ to produce $\{(y_k, z_k)\} \subseteq \mathcal{D}$ such that the distribution of $\{z_k\}$ is more closely aligned with $R \circ \hat{F}_{Z|x}$. Concretely, this is done by forming an empirical CDF of the $\hat{Z}$ samples $\{\hat{z}_j\}$ using binning, re-weighting each bin to match $R \circ \hat{F}_{Z|x}$, then re-sampling from each bin according to the adjusted weights. The full algorithm is summarized in Algorithms 3 and 4: Algorithm 3 describes the procedure for learning the recalibration map R, and Algorithm 4 describes the test time sampling procedure. Crucially, the corresponding $\{y_k\}$ are HDR calibrated.

Proposition 4 *Suppose that $\hat{Z} \sim R \circ \hat{F}_{Z|X}$ and that R is an invertible mapping. Then the distribution of $\hat{Y}$ is HDR calibrated. (proof in Appendix of Chung et al. [2024])*

2.3.5.2 Adjusting the Predictive PDF

The HDR recalibration algorithm from Section 2.3.5.1 produces a predictive distribution (via samples) s.t. the p-HDR contains p proportion of the target observations, on average, $\forall p \in (0, 1)$. However, this predictive distribution can still fail to address dependencies among the output dimensions. This is because, for any fixed x, the HDRs are constructed with *level sets* of $\hat{f}_{Y|x}$, and, if $\hat{f}_{Y|x}$ fails to model dependencies, then the recalibrated samples will also express independence among the output dimensions. We provide an illustration in Figure 2.3. The top row shows that the pre-hoc predictive distribution assumes independence in the output dimensions, which is reflected in the spherical boundaries of the HDRs. After HDR recalibration, the shape of the recalibrated distribution is still spherical, even though the calibration dataset (i.e. ground truth (GT) observations in blue) displays correlation among the dimensions.

This highlights the importance of the projection function $\hat{f}_{Y|X}$, and ideally, $\hat{f}_{Y|X}$ should better reflect the true distribution in order for the recalibration procedure to produce more accurate samples. Further, if we can estimate the errors in $\hat{f}_{Y|X}$ (e.g. correlation, bias) with a held-out dataset, it can be beneficial to adjust $\hat{f}_{Y|X}$ for these factors prior to recalibration.

As a concrete instantiation of this adjustment, we propose a simple procedure to adjust the PDF of multivariate Gaussian distributions by estimating the bias in the predicted mean (i.e. the *location* of the HDRs), standard deviation (i.e. the *width* of the HDRs in each dimension), and the correlation in output dimensions (i.e. the *shape* of the HDRs) with a held-out dataset and correcting the PDF for each of these aspects. We provide details on each adjustment in the Appendix of Chung et al. [2024], and we suggest applying the composition of these adjustments prior to recalibration, as indicated with the ADJUST step in Line 2 of Algorithms 3 and 4. The bottom row of Figure 2.3 provides an illustration of the mean adjustment and correlation adjustment. We can observe that the resulting recalibrated samples more closely reflect the ground truth distribution.

CHAPTER 3

Proposed Work

While the major body of work in UQ is focused on achieving calibration among various UQ metrics, the existing discussion is fairly detached from the actual use-cases of the quantified uncertainties. In fact, there is evidence that calibration may not necessarily even be the ideal notion of "goodness" of a predictive distribution for an application settings that explicitly utilize the uncertainties. One such example is Bayesian optimization (BO), where there are conflicting arguments on the utility of calibration. Deshpande et al. [2024] report the positive utility of calibration for Bayesian optimization (BO), while Foldager et al. [2023] report low correlation between calibration and the performance of BO.

At a high level, we believe that UQ should not be an end goal by itself, and it should demonstrate its value through its use-cases. In the same vein, we believe that achieving good calibration metrics is secondary to the utility the model provides when using it for downstream applications. In that sense, we contend that each application setting or task will define its own set of characteristics that determine the goodness of a predictive distribution, and there is no single metric, *not even calibration*, that universally determines how well a model will perform on all tasks. As a direct example of this, consider the threshold-based decision-making problem setting described in Sahoo et al. [2021a], where an outcome is dictated by whether a target variable of interest, Y is above or below a fixed threshold value, y_0, and an agent takes binary action based on the predicted probability of the target variable being below the threshold, i.e. $\hat{P}(Y \leq y_0)$. As an illustration of the problem setting, suppose an agent has a predictive model which outputs the expected inches of rain for the day (Y), and the agent believe 0.05 inches of rain (y_0) is durable, hence the agent will only decide to take a heavy umbrella if it is likely that it will rain more than 0.05 inches, and the cross product of the binary outcome with binary actions will each incur a loss. Sahoo et al. [2021a] show that predictive distributions that are *average calibrated* will not adequately ensure low loss in this task, and rather a new notion of calibration, which they term *threshold calibration*, is needed. Conversely, it is reasonable to assume that threshold calibration will not necessarily be the optimal measure of the utility that a model will provide for all decision-making problems. Chung et al. [2023] provides a similar discussion on the need for

a problem-specific notion of calibration. Zhao et al. [2021b] provides further examples of how different notions of calibration in *classification* each provide a notion of optimality for different classes of decision-making problems.

There are other instances of problem settings where uncertainty is important, but the discussion of calibration is not clearly elucidated, insignificant, or missing, such as in dynamics learning [Char et al., 2023b, 2021, Mehta et al., 2020], out-of-distribution (OOD) detection [Igoe et al., 2022], or active learning [Settles, 2009].

This is not to discredit calibration as a metric – the ideal predictive distribution will (obviously) achieve perfect calibration and calibration has a unique position as being highly interpretable among metrics for distributional predictions. Rather, we are questioning whether it is *actionable*.

In this thesis, we argue that the purpose of calibration should be largely two-fold:

1. one desires the model outputs to behave like probabilities that describe the outcomes
2. simultaneously, the model outputs should be catered to the task that it will be used for and output meaningful distributional that best aid that task

With these arguments in mind, we propose several research directions that address *using* distributional predictions for downstream applications.

3.1 Calibrated Routing in Soft Mixture of Experts

Mixture of experts (MoE) are a family of model architectures which allow efficient scaling of model size. While the standard deep learning architecture features dense layers, the weights of which are activated by all inputs, in MoEs, these layers are divided up into *multiple* layers, and each input unit (i.e. token) will selectively activate one or a subset of these layers. This allows one to increase the total number of parameters in a model while keeping the computational cost of a single forward pass manageable.

While the archetypical MoE is the "Sparse MoE", which discretely routes each input token to a subset of the experts, the discrete routing operation causes various issues in optimization and inference. To alleviate these issues, the recently introduced Soft MoE [Puigcerver et al., 2024] eschews discrete matching in favor of a smoother approach. It computes for each expert a convex combination of the input tokens, and the expert only sees this convex combination. The final output of the model is then a convex combination of each expert's output. This approach is fully differentiable, and hence is more stable than the Sparse MoE. This novel Soft MoE architecture has been shown to outperform all other baselines on challenging large scale vision tasks, and can scale to thousands of experts [Puigcerver et al., 2024]. Moreover, recent results show that the

Soft MoE is a promising avenue towards providing empirical scaling laws for deep reinforcement learning [Obando Ceron et al., 2024].

We briefly discuss the Soft MoE architecture [Puigcerver et al., 2024]. Let $X \in \mathbb{R}^{m \times d}$ denote the tokenized input, so that there are m tokens each in $\mathbb{R}^d$. The MoE layer is equipped with n experts $\{f_j : \mathbb{R}^d \to \mathbb{R}^d\}_{j=1}^n$, each of which is typically implemented as a feedforward network. The router is parameterized by $\Phi \in \mathbb{R}^{d \times n}$. Given an input X, the parameters Φ are used to compute matrices $D(X), C(X) \in \mathbb{R}^{m \times n}$ which are defined elementwise as

$$D(X)_{ij} = \frac{\exp\left((X\Phi)_{ij}\right)}{\sum_{i'=1}^m \exp\left((X\Phi)_{i'j}\right)} \quad \text{and} \quad C(X)_{ij} = \frac{\exp\left((X\Phi)_{ij}\right)}{\sum_{j'=1}^n \exp\left((X\Phi)_{ij'}\right)}. \tag{3.1}$$

Note that each column of $D(X)$ and each row of $C(X)$ sums to one. With this notation in hand, we formally define the Soft MoE layer below.

Definition 1 *The Soft MoE is a function* $\mathrm{sMoE}^{\Phi}_{\{f_j\}_{j=1}^n} : \mathbb{R}^{m \times d} \to \mathbb{R}^{m \times d}$ *defined as*

$$\mathrm{sMoE}^{\Phi}_{\{f_j\}_{j=1}^n}(X) = C(X)\widetilde{Y}(X) \quad \textit{where} \quad \widetilde{Y}(X) = \begin{bmatrix} f_1\left((D(X)^T X)_1\right) \\ \vdots \\ f_n\left((D(X)^T X)_n\right) \end{bmatrix}.$$

The Soft MoE thus computes n different convex combinations of the tokens in X, where the weights of the jth convex combination are given by the jth column of $D(X)$. It then applies expert f_j to the jth convex combination, for each $j = 1, 2 \ldots n$. Finally, it computes m different convex combinations of these expert outputs, where the weights of the ith convex combination are given by the ith row of $C(X)$. Note that each expert processes a single vector in $\mathbb{R}^d$, and that sMoE is differentiable whenever the experts are. This results in more stable training relative to Sparse MoE, where each expert is given a subset of the m tokens via a discrete matching algorithm. The Soft MoE has shown significant empirical success in vision [Puigcerver et al., 2024] and reinforcement learning [Obando Ceron et al., 2024].

We note that the Soft MoE module leverages 2 softmax distributions: $D(X)$ and $C(X)$. The i$^{\text{th}}$column of $D(X)$ denotes the distribution over the input tokens that the i$^{\text{th}}$expert should attend to, and the i$^{\text{th}}$row of $C(X)$ denotes the distribution over the expert outputs that the i$^{\text{th}}$output token should attend to.

We ask the question of whether these distributions are *calibrated*. In the context of model inference for prediction, we can define the task at hand as accurate classification (i.e. aiming for test accuracy), and with this notion of a task, we can define the purpose of calibration of the distributions $C(X)$ and $D(X)$ as making them as informative as possible to produce accurate

predictions.

Further, if one could efficiently identify the most informative token/expert to attend to based on the calibrated distribution, this information can be used to prune the model during inference and save compute.

3.2 Parameterized Proper Scoring Rules

Proper scoring rules are functions which assess the quality of predicted probability distributions. Given a distributional prediction P that is an element of the space of probability distributions Δ, and an outcome space $\mathcal{Y}$, a proper scoring rule $S : \Delta \times \mathcal{Y} \to \mathbb{R}$ is a function that evaluates the quality of the prediction P based on the observed outcome. Given an outcome y, the assigned score is $S(P, y)$, and when the $y \sim Q$, the expected score is denoted as $S(P, Q)$.

Proper scoring rules enjoy the property that the expected score is maximized by the true distribution of the outcomes, i.e. $S(Q, Q) \geq S(P, Q), \forall P \in \Delta$, and the score is said to be *strictly* proper if the maximum value is attained if and only if $P = Q$.

Gneiting and Raftery [2007] summarized various properties of proper scoring rules that have been discussed across various works, and they make note of the correspondence between proper scoring rules and convex functions. Specifically, each proper scoring rule S corresponds to a convex function $G : \Delta \to \mathbb{R}$, whereby for any predicted probability P and outcome y,

$$S(p, y) = G(p) - \mathbb{E}_{w \sim p} G'(p, w) + G'(p, y), \tag{3.2}$$

where $G'(P, w)$ denotes the subtangent of G at $P \in \Delta$, evaluated at the point $w \in \mathcal{Y}$.

For simplicity, we confine the problem setting to the classification with m total classes and Δ is the $m - 1$ probability simplex, i.e. P is an m-dimensional vector denoting class probabilities, and $G'(P)$ is also an m-dimensional vector. Then we can rewrite Eq. 3.2 as

$$S(p, y) = G(p) - G'(p)^T p + G'(p)_y, \tag{3.3}$$

where $G'(P)_y$ denotes the y^{th}element of $G'(P)$ (Theorem 2, Gneiting and Raftery [2007]; they also refers to this as the *Savage representation* of proper scoring rules.)

Given the ground truth distribution Q, the expected score, S(P, Q), is then,

$$S(P,Q) = \mathbb{E}_{y\sim Q}[G(P) - G'(P)^T P + G'(P)_y] \tag{3.4}$$
$$= G(P) - G'(P)^T P + \mathbb{E}_{y\sim Q}[G'(P)_y] \tag{3.5}$$
$$= G(P) - G'(P)^T P + G'(P)^T Q \tag{3.6}$$
$$= G(P) - G'(P)^T (P - Q). \tag{3.7}$$
$$\tag{3.8}$$

Further, the maximum score achievable is $S(Q, Q)$,

$$S(Q,Q) = G(Q) - G'(Q)^T (Q - Q) \tag{3.9}$$
$$= G(Q). \tag{3.10}$$
$$\tag{3.11}$$

There are many interpretations of the function G. Gneiting and Raftery [2007] refers to this function as

- *information measure*,
- *generalized entropy function*,
- or simply, *entropy function*.

For example, when G is the negative Shannon Entropy function $G(P) = -\sum_{i\in[m]} P_i \log p_i$, the corresponding scoring rule is the log score, also known as the cross-entropy loss, $S(P, y) = \log P_y$.

An alternative viewpoint of G is that it is a

- *utility function*

in a decision-making setting. Suppose an agent makes probabilistic predictions about the outcome, and always takes the Bayes decision rule according to the belief: given a the probability P, denote the Bayes decision rule as a_P, and the utility function as $U : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$. By definition of Bayes decision rule,

$$a_P = \arg\sup_{a\in\mathcal{A}} \mathbb{E}_{y\sim P}[U(y, a)] \tag{3.12}$$

If we denote

$$S(P, y) = U(y, a_P), \tag{3.13}$$

i.e., $S(P, y)$ is the maximal utility the agent can incur under their belief, then

$$G(Q) = S(Q, Q) \tag{3.14}$$
$$= \mathbb{E}_{y\sim Q}[S(Q, y)] \tag{3.15}$$
$$= \mathbb{E}_{y\sim Q}U(y, a_Q), \tag{3.16}$$

which is the maximum utility the agent can derive from the decision-making task where the ground truth distribution of outcomes is Q.

Given these interpretations, we ask the following questions:

- *There is evidence of the squared-loss performing at least as well as cross-entropy for a wide-range of classification tasks [Hui and Belkin, 2020] - can we devise better loss functions that are specific to how the model will be used downstream?*
- *For these cases, what interpretation would the utility function (or entropy) have? Would it suggest an better alternative utility function compared to the Shannon entropy function?*

Furthermore, we know from Eq. 3.2 that *any* convex function produces a proper scoring rule. Crucially, we can *parameterize* the convex function and aim to *learn* the optimal convex function for the problem setting at hand.

Consider the utility function G as follows:

$$G(p) = p^T A \log p \tag{3.17}$$

where A is an $m \times m$ matrix and $\log p$ denotes the element-wise logarithm of p. While this function is not convex for any arbitrary A, it should be convex under suitable conditions. For example, the Shannon entropy function is subsumed under this parameterization when $A = I$. Therefore, using the cross-entropy loss signifies using the above utility function with A confined to be the identity matrix.

3.2.1 Learnable Utility Functions

As my first proposed work, I aim to work with the conjecture that, the optimal utility function is different for each downstream task and that it is learnable. I.e. depending on *how* the classification model will be used in the end, it can be advantageous to deviate from Shannon entropy as the utility function and 1) set a different, designated convex function as the utility function and 2) derive the proper scoring rule implied by the utility function and optimize it during training. We can rely on the fact that *any* convex function produces a proper scoring rule, which by definition, is optimized at the true underlying distribution.

There are many different ways to parameterize a convex function. One simple form is the quadratic:

$$G_\theta(p) = p^T A p \tag{3.18}$$

with $\theta = \{A\}$ and A a PSD matrix.

Other parameterizations include

$$G_\theta(p) = p^T A \log p \tag{3.19}$$

where A is a diagonal matrix and

$$G_\theta(p) = \sum_{i \in [n]} \phi_i(p) \tag{3.20}$$

where each $\{\phi_i\}$ are convex functions.

A much more complex form is using input convex neural networks [Amos et al., 2017]

$$G_\theta(p) = \texttt{ICNN}_\theta(p) \tag{3.21}$$

In the first line of proposed work, we aim to formalize an algorithm for learning the function G. At the time of writing, it is unclear what the objective for learning G should be. Given that the status quo is to use the negative Shannon entropy function as the utility function guiding optimization (via the cross-entropy loss) in virtually all classification problems, it stands to reason that given different downstream applications of the classification model, a different utility function (and its derived proper scoring rule) can provide higher downstream performance.

We believe there are various learning schemes that are possible:

- learning from the loss/utility function that defines the downstream application
- learning from data (i.e. samples from the ground truth distribution)

3.2.2 Application to Language Models

We believe language models provide for a unique and impactful application of the techniques proposed above. Lanaguage models are typically modeled as predicting future tokens, each of which are considered discrete classes from a dictionary which define the vocabulary set. In modern architectures, language modeling presents perhaps the largest scale classification problem yet. E.g. Llama 3 [Dubey et al., 2024], the tokenizer utilizes 128,000 tokens, meaning the prediction task is a classification task among 128K classes. The standard practice is to use the cross-entropy loss during training. Again, using the cross-entropy loss indicates using negative Shannon entropy as the fixed utility function, which is symmetric in all classes (i.e. tokens).

We believe there are several interesting research directions when applying the proposed techniques.

3.2.2.1 Next token prediction

We consider the pre-training objective, which is the next token prediction task. This setting is analogous to the standard supervised learning for classification. The cross-entropy loss treats each token symmetrically, and this also reflected in the symmetry of the Shannon entropy function. However, we conjecture that there are tokens which are more important or informative than others. We hypothesize that encoding this asymmetry in the utility function, and subsequently the proper scoring rule as the loss function, should help accelerate training and convergence to a better local optimum. There are several questions and challenges in this direction.

1. Operating in the semantic space of tokens: While benchmark datasets in classification such as MNIST or CIFAR feature independent classes, the classes in language modeling are tokens which, even at the atomic token level, should display dependence. When considering the semantic meaning of words/sentences that each token often comprise, their dependence should be even starker. The utility function should ideally model these dependencies.

2. High dimensional probability vectors: In addition, the predicted distribution when predicting a next token is now 128K dimensional. If we consider the quadratic parameterization in Eq. 3.18, the matrix A becomes 128K$\times$128K $\approx$ 16B dimensional. One could use low-rank approximations of the parameters to alleviate this burden, or use ICNN's with small hidden layers.

3.2.2.2 Fine tuning

More often than not, language models are adapted for a specific use-case via a fine-tuning process that is also referred to as "alignment". It would be an interesting direction to encode the desired, aligned behavior into a utility function, which in turn would provide for a proper scoring rule which one can optimize the language model with. This would necessitate several extensions to the framework described in the previous section.

1. Sentence-level utility and scoring functions: Both the functions G and S operate on a single probability vector that predict a single token, but it would be ideal to devise a sentence-level function which can operate on inputs that is comprised of a sequence of tokens.

2. Alignment of the utility with the dataset: The alignment dataset already encodes a notion of utility: given a certain prompt, the "winning" response is preferred over the "losing" response. The utility function should ideally align with this implied utility in the data.

We intend to follow standard evaluation protocol in alignment to evaluate the proposed method, e.g. [Rafailov et al., 2024, Meng et al., 2024].

3.3 Timeline

Figure 3.1 shows my plan for a timeline moving forward.

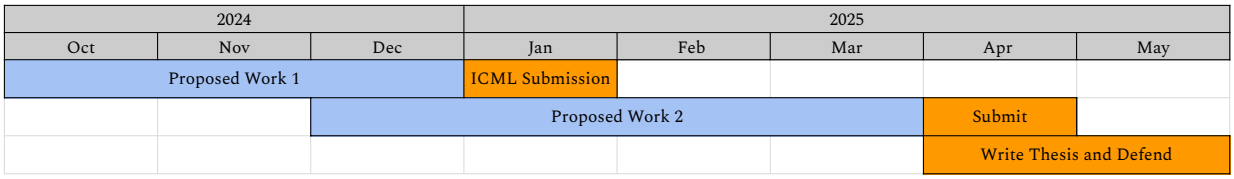


Figure 3.1: **Timeline for graduation.**

BIBLIOGRAPHY

J. Abbate, R. Conlin, and E. Kolemen. Data-driven profile prediction for diii-d. *Nuclear Fusion*, 61 (4):046027, 2021.

B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *International conference on machine learning*, pages 146–155. PMLR, 2017.

R. Askanazi, F. X. Diebold, F. Schorfheide, and M. Shin. On the comparison of interval forecasts. *Journal of Time Series Analysis*, 39(6):953–965, 2018.

P. Barbe, C. Genest, K. Ghoudi, and B. Remillard. On kendall's process. *journal of multivariate analysis*, 58(2):197–229, 1996.

A. Belloni and R. L. Winkler. On multivariate quantiles under partial ordering. Technical report, 2009.

V. Bowman, D. Silk, U. Dalrymple, and D. Woods. Uncertainty quantification for epidemiological forecasts of covid-19 through combinations of model predictions. *arXiv preprint arXiv:2006.10714*, 2020.

J. Bracher, E. L. Ray, T. Gneiting, and N. G. Reich. Evaluating epidemic forecasts in an interval format. *PLoS computational biology*, 17(2):e1008618, 2021.

A. J. Cannon. Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic environmental research and risk assessment*, 32(11):3207–3225, 2018.

I. Char, Y. Chung, M. Boyer, E. Kolemen, and J. Schneider. A model-based reinforcement learning approach for beta control. In *APS Division of Plasma Physics Meeting Abstracts*, volume 2021, pages PP11–150, 2021.

I. Char, J. Abbate, L. Bardóczi, M. Boyer, Y. Chung, R. Conlin, K. Erickson, V. Mehta, N. Richner, E. Kolemen, et al. Offline model-based reinforcement learning for tokamak control. In *Learning for Dynamics and Control Conference*, pages 1357–1372. PMLR, 2023a.

I. Char, Y. Chung, R. Shah, W. Neiswanger, and J. Schneider. Correlated trajectory uncertainty for adaptive sequential decision making. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023b.

I. Char, Y. Chung, J. Abbate, E. Kolemen, and J. Schneider. Full shot predictions for the diii-d tokamak via deep recurrent networks. *arXiv preprint arXiv:2404.12416*, 2024.

K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Y. Chung, W. Neiswanger, I. Char, and J. Schneider. Beyond pinball loss: Quantile methods for calibrated uncertainty quantification. *Advances in Neural Information Processing Systems*, 34: 10971–10984, 2021.

Y. Chung, A. Rumack, and C. Gupta. Parity calibration. In *Uncertainty in Artificial Intelligence*, pages 413–423. PMLR, 2023.

Y. Chung, I. Char, and J. Schneider. Sampling-based multi-dimensional recalibration. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 21–27 Jul 2024.

P. Cui, W. Hu, and J. Zhu. Calibrated reliable regression using maximum mean discrepancy. *Advances in Neural Information Processing Systems*, 33, 2020.

N. Dalmasso, T. Pospisil, A. B. Lee, R. Izbicki, P. E. Freeman, and A. I. Malz. Conditional density estimation tools in python and r with applications to photometric redshifts and likelihood-free cosmological inference. *Astronomy and Computing*, 30:100362, 2020.

S. Deshpande, C. Marx, and V. Kuleshov. Online calibrated and conformal prediction improves bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1450–1458. PMLR, 2024.

N. S. Detlefsen, M. Jørgensen, and S. Hauberg. Reliable training and estimation of variance networks. *arXiv preprint arXiv:1906.03260*, 2019.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual review of fluid mechanics*, 51:357–377, 2019.

M. Fasiolo, S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude. Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, pages 1–11, 2020.

J. Foldager, M. Jordahn, L. K. Hansen, and M. R. Andersen. On the role of model uncertainties in bayesian optimisation. In *Uncertainty in Artificial Intelligence*, pages 592–601. PMLR, 2023.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

L. G. Galvão and M. N. Huda. Pedestrian and vehicle behaviour prediction in autonomous vehicle system — a review. *Expert Systems with Applications*, 238:121983, 2024. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.121983. URL https://www.sciencedirect.com/science/article/pii/S0957417423024855.

C. Genest and L.-P. Rivest. On the multivariate probability integral transformation. *Statistics & probability letters*, 53(4):391–399, 2001.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

C. Gupta and A. Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In *International Conference on Machine Learning*, pages 3942–3952. PMLR, 2021.

D. Harrison, D. Sutton, P. Carvalho, and M. Hobson. Validation of bayesian posterior distributions using a multidimensional kolmogorov–smirnov test. *Monthly Notices of the Royal Astronomical Society*, 451(3):2610–2624, 2015.

U. Hébert-Johnson, M. P. Kim, O. Reingold, and G. N. Rothblum. Calibration for the (computationally-identifiable) masses. *arXiv preprint arXiv:1711.08513*, 2017.

M. P. Holmes, A. G. Gray, and C. L. Isbell. Fast nonparametric conditional density estimation. *arXiv preprint arXiv:1206.5278*, 2012.

L. Hui and M. Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

R. J. Hyndman. Computing and graphing highest density regions. *The American Statistician*, 50 (2):120–126, 1996.

R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4):315–336, 1996.

C. Igoe, Y. Chung, I. Char, and J. Schneider. How useful are gradients for ood detection really? *arXiv preprint arXiv:2205.10439*, 2022.

R. Izbicki, G. Shimizu, and R. B. Stern. Cd-split and hpd-split: Efficient conformal regions in high dimensions. *The Journal of Machine Learning Research*, 23(1):3772–3803, 2022.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE access*, 6:36218–36234, 2018.

R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

V. Kuleshov and S. Deshpande. Calibrated and sharp uncertainties in deep learning via density estimation. In *International Conference on Machine Learning*, pages 11683–11693. PMLR, 2022.

V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*, 2018.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

K. Maciejowska, J. Nowotarski, and R. Weron. Probabilistic forecasting of electricity spot prices using factor quantile regression averaging. *International Journal of Forecasting*, 32(3):957–965, 2016.

A. Malik, V. Kuleshov, J. Song, D. Nemer, H. Seymour, and S. Ermon. Calibrated model-based deep reinforcement learning. *arXiv preprint arXiv:1906.08312*, 2019.

V. Mehta, I. Char, W. Neiswanger, Y. Chung, A. O. Nelson, M. D. Boyer, E. Kolemen, and J. Schneider. Neural dynamical systems. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Y. Meng, M. Xia, and D. Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

A. H. Murphy. A new vector partition of the probability score. *Journal of applied Meteorology*, 12 (4):595–600, 1973.

R. B. Nelsen, J. J. Quesada-Molina, J. A. Rodríguez-Lallena, and M. Úbeda-Flores. Kendall distribution functions. *Statistics & probability letters*, 65(3):263–268, 2003.

J. S. Obando Ceron, G. Sokar, T. Willi, C. Lyle, J. Farebrother, J. N. Foerster, G. K. Dziugaite, D. Precup, and P. S. Castro. Mixtures of experts unlock parameter scaling for deep RL. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38520–38540. PMLR, 21–27 Jul 2024.

T. Pearce, F. Leibfried, A. Brintrup, M. Zaki, and A. Neely. Uncertainty in neural networks: Approximately bayesian ensembling. *arXiv preprint arXiv:1810.05546*, 2018a.

T. Pearce, M. Zaki, A. Brintrup, and A. Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *arXiv preprint arXiv:1802.07167*, 2018b.

J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

J. Puigcerver, C. R. Ruiz, B. Mustafa, and N. Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024.

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

F. Rodrigues and F. C. Pereira. Beyond expectation: deep joint mean and quantile regression for spatiotemporal problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

R. Sahoo, S. Zhao, A. Chen, and S. Ermon. Reliable decisions with threshold calibration. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1831–1844. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0e65972dce68dad4d52d063967f0a705-Paper.pdf.

R. Sahoo, S. Zhao, A. Chen, and S. Ermon. Reliable decisions with threshold calibration. *Advances in Neural Information Processing Systems*, 34:1831–1844, 2021b.

B. Settles. Active learning literature survey. 2009.

D. M. Sexton, J. M. Murphy, M. Collins, and M. J. Webb. Multivariate probabilistic projections using imperfect climate models part i: outline of methodology. *Climate dynamics*, 38:2513–2542, 2012.

H. Song, T. Diethe, M. Kull, and P. Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. PMLR, 2019.

W. Stute et al. On almost sure convergence of conditional empirical distribution functions. *The Annals of Probability*, 14(3):891–901, 1986.

N. Tagasovska and D. Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pages 6414–6425, 2019.

K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing, and Z. W. Ulissi. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1(2):025006, 2020.

R. L. Winkler. A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972.

Q. Xu, K. Deng, C. Jiang, F. Sun, and X. Huang. Composite quantile regression neural network with applications. *Expert Systems with Applications*, 76:129–139, 2017.

T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001.

B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

D. Zhao, N. Dalmasso, R. Izbicki, and A. B. Lee. Diagnostics for conditional density models and bayesian inference algorithms. In *Uncertainty in Artificial Intelligence*, pages 1830–1840. PMLR, 2021a.

S. Zhao, T. Ma, and S. Ermon. Individual calibration with randomized forecasting. In *International Conference on Machine Learning*, pages 11387–11397. PMLR, 2020.

S. Zhao, M. Kim, R. Sahoo, T. Ma, and S. Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34: 22313–22324, 2021b.

J. F. Ziegel and T. Gneiting. Copula calibration. *Electronic journal of statistics*, 8(2):2619–2638, 2014.