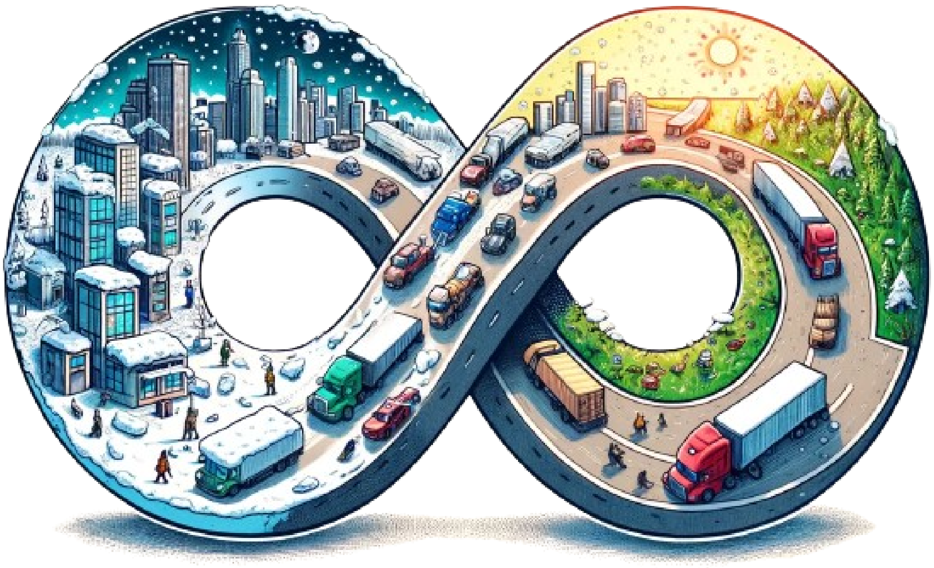
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG



Probabilistic Calibration in a Few-Shot Domain Adaptation Setting

Assessing machine learning models' characteristics in bridging covariate shifts from new operational conditions for electric trucks

Master's thesis in Mathematical Sciences

Oscar Adamsson
Jonas Röst

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Probabilistic Calibration in a Few-Shot Domain Adaptation Setting

Assessing machine learning models' characteristics in bridging covariate shifts from new operational conditions for electric trucks

Oscar Adamsson
Jonas Röst

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Probabilistic Calibration in a Few-Shot Domain Adaptation Setting
Assessing machine learning models' characteristics in bridging covariate shifts from new operational conditions for electric trucks
Oscar Adamsson
Jonas Röst

Supervisor: Professor Klas Modin, Mathematical Sciences
Advisors: Jenny Eriksson, Einride; Ana Fotina, Einride; Claes Andersson, Einride
Examiner: Professor Klas Modin, Mathematical Sciences

Master's Thesis 2024
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Probabilistic Calibration in a Few-Shot Domain Adaptation Setting
Assessing machine learning models' characteristics in bridging covariate shifts from new operational conditions for electric trucks
Oscar Adamsson
Jonas Röst
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis investigates the application of probabilistic machine learning models within a Few-Shot Domain Adaptation (FSDA) setting to address *covariate shifts* induced by new operational conditions for electric trucks. By leveraging data-driven methods instead of the truck's physical properties, the thesis assesses the characteristics and robustness of different machine learning models in predicting energy consumption under various new conditions. The study focuses on scenarios involving uni-, bi-, and multivariate covariate shifts posed by colder temperatures, a new route type, and a new vehicle manufacturer. Utilizing real-world data from electric truck drives, the models are trained on source data, adapted using limited target domain data, and assessed for their probabilistic calibration in the target domain. The findings indicate that out of the two baseline models, Ridge regression models, the *source-trained baseline model* performs well under simpler shifts but struggles with multivariate shifts where the *target-only baseline model* excels given sufficient target domain data. Hierarchical Bayesian linear regression shows high adaptability when covariate shift affects hierarchical levels of the model. Gaussian process regression improves comparatively well with adaptation. However, the results indicate a possible sensitivity to kernel selection. Bayesian neural networks face challenges with prediction mean accuracy and high sensitivity to individual samples, further research is needed to determine the model's feasibility in a FSDA setting. These insights provide valuable guidance for fleet management companies in improving decision-making and operational efficiency under new driving conditions through accurate probabilistic energy consumption modeling.

Keywords: few-shot domain adaptation, probabilistic machine learning models, covariate shift, distribution shift, energy consumption prediction, probabilistic calibration, predictive distributions, electric trucks.

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Professor Klas Modin, for his guidance, support, and encouragement throughout this thesis. We are also deeply thankful to our advisors from Einride: Jenny Eriksson, Ana Fotina, and Claes Andersson, for their insightful feedback and practical perspectives, which significantly enriched this work.

Oscar Adamsson & Jonas Röst, Gothenburg, 2024-06-10

Contents

List of Figures

List of Tables

1

Introduction

The shift towards the electrification of transport vehicles is transforming the landscape of fleet logistics. This transition, particularly evident in the adoption of electric trucks, increases the demands on fleet managers. The challenges stem from the relatively sparse charging infrastructure and the extended time required for recharging compared to Internal Combustion Engine (ICE) trucks, both of which complicate the operations. In addition, maintaining electric vehicle (EV) batteries within specific charge intervals is crucial to avoiding rapid depreciation of the battery and ensuring their sustained performance. The increased complexities of managing fleets of electric trucks underscore the need for robust energy consumption modeling to manage the fleet effectively.

Electric truck manufacturers disclose information about the energy consumption of their trucks. However, this information falls short when relied on in isolation. The uncertainty of these estimates, for various driving conditions, is seldom provided and can consequently lead to misguided confidence when, for example, planning the fleet operations. This lack of quality estimates for truck energy consumption underscores the need for data-driven methods, that are able to quantify the uncertainty, based on empirical data.

From a strategic standpoint, fleet managers within the transport sector seeking to leverage data-driven methods, face a substantial challenge: varying conditions lead to diverse data, each stemming from distinct data-generating processes or distributions. The degree of difference between two disparate environments, e.g., routes, can significantly impact the performance of predictive models [1]. This phenomenon between training and test data is known as *distributional shift*, and is a common challenge for data-driven models applied in real-world scenarios. These shifts are particularly pronounced for transport operators when planning for new routes, where the conditions are different, or when introducing additional truck models. For a fleet manager seeking to expand operations by acquiring new customers while maintaining operational efficiency, making accurate energy consumption predictions — and consequently addressing distributional shifts — is pivotal.

In a real-world setting, data for previously unobserved driving conditions can often be collected in limited amount, by making *test-drives* in those conditions. Leveraging this test-drive data presents an opportunity to reduce the predictive errors caused by distributional shift from unobserved environments. Thus, enabling improved energy consumption modeling.

Traditionally, data-driven models in the field of energy consumption estimation produce point predictions without accounting for the uncertainty of those predictions [2]. Probabilistic modeling, however, offers a more nuanced approach by predicting probability distributions over estimates rather than single-point values. These more informative predictions can significantly enhance the application of energy consumption modeling for electric trucks. Probabilistic estimates can be interpreted as point predictions through the expectation of their distributions. Additionally, the full probability distribution provides valuable insights for more informed decision-making.

Research into energy consumption modeling of electric trucks remains limited [2]. Historically, the research focus has been predominantly on micro-scale models, which are designed for controlling the electric trucks' immediate energy use rather than on developing macro-scale models that could support both strategic and operational fleet management decision-making. In addition, deriving causal structures from the physical properties of the electric trucks' has been the most common technique researched [3]. While these approaches offer valuable insights, they differ significantly from the data-driven methodologies that have proven effective in modeling energy consumption for personal electric vehicles [4]–[12].

In summary, a probabilistic energy consumption model that can swiftly adjust to new conditions, based on limited data from that environment, would significantly improve strategic planning capabilities, enable more efficient fleet management and well-informed decision making, when entering new customer geographies or incorporating new truck models.

1.1 Problem

As outlined in the introduction, the transition towards electric trucks demands accurate energy consumption modeling in unobserved environments with new conditions. In addition, the ability to quantify the uncertainty of predictions can enable more well-informed decision-making. Several obstacles hinder this goal.

Firstly, environments with new driving conditions often induce *covariate shift*, a type of distributional shift, in the scenarios covered in this thesis. Secondly, estimating a probability distribution instead of a deterministic point prediction enhances the practical utility of the model but also introduces more complexity and assumptions about the underlying data. Lastly, the amount of data collectable, test-drives, from the environment with new conditions is limited and therefore, intuitively, modeling on that data in isolation can result in low degree of robustness, yet the data naturally posits valuable information for the task at hand.

Suppose an environment with one type of driving conditions is defined as the **source domain** ($\mathcal{S}$). Further, suppose another environment with different, new conditions is defined as the **target domain** ($\mathcal{T}$). Let S and T_{adapt} be two sets of samples, representing the data collected from each domain, respectively. The academic challenges posed can then be expressed accordingly:

Covariate Shift: $p_{\mathcal{S}}(\boldsymbol{x}) \neq p_{\mathcal{T}}(\boldsymbol{x}) \ \wedge \ p_{\mathcal{S}}(y|\boldsymbol{x}) = p_{\mathcal{T}}(y|\boldsymbol{x})$

The feature distributions of the source domain, $\mathcal{S}$, and target domain, $\mathcal{T}$, are not aligned. However, the dependency between the driving conditions, $\boldsymbol{x}$, and the energy consumption, y, is consistent across the two domains.

Probabilistic Prediction: $y \sim p(y|\boldsymbol{x})$

The label, energy consumption, is modeled as a conditional probability distribution of the features, driving conditions.

Few labeled target domain samples: $N_{\mathcal{S}} \gg N_{\mathrm{T}_{\mathrm{adapt}}}$

There are few labeled samples available from the target domain, $N_{\mathrm{T}_{\mathrm{adapt}}} = |T_{\mathrm{adapt}}|$. There are substantially more samples from the source domain, $N_{\mathrm{S}} = |S|$.

Ideally, a model trained on a source domain would learn the true conditional distribution, $p_{\mathcal{S}}(y|\boldsymbol{x})$, and consequently generalize well across other domains with the same task, $p_{\mathcal{T}}(y|\boldsymbol{x}) = p_{\mathcal{S}}(y|\boldsymbol{x})$. However, machine learning (ML) models are subject to *bias*, from the choice of hypothesis class, and *estimation error* from the limited number of observations available. Consequently, the trained model, $f(\boldsymbol{x}; \theta_{\mathrm{S}}) \rightarrow \hat{p}(y|\boldsymbol{x})$, is often overfitted to the distribution of the source domain data, S. If the target and source domain data are independently and identically distributed (i.i.d), this does not lead to significant performance loss since the target domain data is similarly distributed to the source domain data that the model has been trained on. However, the source trained model's deviation from the true representation becomes apparent under covariate shift, where the task remains identical but the conditions change, resulting in a significant reduction in performance.

The paradigm of Domain Adaptation (DA) is a well-researched area aimed at addressing the challenges posed by distributional shift, including covariate shift, using various techniques. In our case, where there is a limited amount of labeled samples from the target domain, the scope can be further constrained to Few-Shot Domain Adaptation (FSDA). Given the need for probabilistic predictions, our problem can specifically be categorized as FSDA within a probabilistic setting.

1.2 Aim

This thesis aims to assess the characteristics and efficacy of various machine learning models in addressing covariate shift arising from new operational conditions for electric trucks. Specifically, it evaluates the robustness of these models in providing accurate probabilistic predictions for new driving conditions, given different quantities of test-drives in those conditions.

Mathematically, the thesis sets out to assess the models, $f(\boldsymbol{x}; \theta)$, by evaluating the probabilistic accuracy of their predictive distributions, $\hat{p}(y|\boldsymbol{x})$, in the target domain. Consequently, to which degree Equation 1.1 holds:

$$p_{\mathcal{T}}(y|\boldsymbol{x}) \approx \hat{p}(y|\boldsymbol{x}) \tag{1.1}$$

where the predictions, by the Few-Shot Domain Adaptation context, are generated accordingly:

$$\hat{p}(y|\boldsymbol{x}) = f(\boldsymbol{x}; \theta_{\mathrm{S \cup T_{adapt}}}) \tag{1.2}$$

and subject to the three conditions posed by the challenges (e.g., covariate shift) described in the Problem (Section 1.1).

1.3 Delimitations

To maintain a clear and manageable scope, some key delimitations have been made. Firstly, the thesis explores data-driven methods for energy consumption modeling rather than relying on predefined physical properties of energy consumption for electric trucks. Secondly, the analysis centers on evaluating the performance of these models in the target domain. The study does not consider the performance of the models on the source domain before or after adaptation. Hence, taking no account to the models ability to generalize across the two. Lastly, the study is confined to typical operational conditions and routes relevant to the fleet management company's activities, excluding unrealistic or atypical driving conditions. In addition, the study does not account for long-term changes in energy consumption patterns.

2

Theory

In the following sections, theory relevant to the thesis will be presented. First, probabilistic regression and various machine learning models will be discussed in Section 2.1, categorized by the paradigms of Frequentism and Bayesianism. Thereafter, in Section 2.2, techniques to evaluate probabilistic regression models will be described. Lastly, theory on the phenomenon of *distributional shift* will be given in Section 2.3.

2.1 Probabilistic Regression

Probabilistic regression diverges from deterministic models by estimating distributions over outcomes, thereby quantifying prediction uncertainty. This approach involves estimating the conditional distribution, $Y|X \sim p_{Y|X}$, for *discriminative* models rather than providing deterministic point estimates, $Y|X = E[p(Y|X)]$. *Generative* models can be employed as well, aiming to estimate the joint distribution $Y, X \sim p_{Y,X}$.

In supervised machine learning, this methodology can be divided into Bayesian and Frequentist schools of thought. Bayesian methods incorporate prior knowledge and update beliefs about model parameters by observing data, resulting in a posterior distribution $p(\theta|D)$ using Bayes' theorem. Frequentist methods, while not inherently probabilistic but predominantly associated with deterministic point estimates, can be modified to estimate prediction uncertainty through various techniques. However, what inherently sets the two apart is whether the model parameters, θ, are assumed deterministic (Frequentist approach) or random (Bayesian approach).

2.1.1 Frequentist Approach

The Frequentist approach considers parameters of interest, θ, as fixed but unknown quantities, estimable from observed data. In supervised learning, most models are discriminative and are developed by approximating the true conditional probability distribution of the target variable given the features, $p(y|\boldsymbol{x})$, directly from the training data, D. This approximation, denoted $\hat{p}_D(y|\boldsymbol{x})$, is used to inform predictions. Specifically, the prediction $\hat{y}(\boldsymbol{x})$ at a new point $\boldsymbol{x}$ is the one that minimizes the expected loss with respect to $\hat{p}(y|\boldsymbol{x})$, expressed mathematically as:

$$\hat{y}(\boldsymbol{x}) = \underset{\hat{y}}{\operatorname{argmin}} \mathbb{E}_{y\sim\hat{p}_D(y|\boldsymbol{x})}[\ell(y, \hat{y})|\boldsymbol{x}], \tag{2.1}$$

where $\ell(y, \hat{y})$ is the loss function. Using empirical data, the Frequentist approach estimates the fixed parameters, θ, of the predictive function, $\hat{y}$. Maximum Likelihood Estimation (MLE) is used for parameter estimation,

$$\theta_{\text{MLE}} = (\boldsymbol{w}_{\text{MLE}}, \beta_{\text{MLE}}) = \arg\max_{\boldsymbol{w},\beta} \hat{p}(t|\boldsymbol{x}_D, \boldsymbol{w}, \beta)$$

where $(\boldsymbol{w}, \beta)$ are the weights and error variance respectively, both attained by learning from the observed data, D, because Frequentism assumes the model parameters to be deterministic.

2.1.1.1 Linear Regression

In the case of linear regression, using a quadratic loss function and parameters learned from the data (MLE), Equation 2.1 can be simplified to:

$$\hat{y}(\boldsymbol{x}) = \mathbb{E}_{y\sim p(y|\boldsymbol{x},\theta_{\text{MLE}})}[y|\boldsymbol{x}] = \mu(\boldsymbol{x}, \boldsymbol{w}_{\text{MLE}}) = w_0 + \sum_{j=1}^{M} w_j x_j$$

The predictive function, $\hat{y}$, provides a point prediction with no associated uncertainty. However, the uncertainty can be included if instead expressed accordingly:

$$\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x} + \epsilon = \mu(\boldsymbol{x}, \boldsymbol{w}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \beta^{-1})$$

Equivalently:

$$\hat{p}(y|\boldsymbol{x}, \boldsymbol{w}, \beta) = p(y|\boldsymbol{x}, \theta) = \mathcal{N}(y|\mu(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}), \quad \theta = (\boldsymbol{w}, \beta)$$

This formulation, though probabilistic, remains in the Frequentist realm. It accounts for the noise in the data by incorporating an error term, ϵ, modelling *aleatory uncertainty*, but assumes fixed unknown parameters and by that neglects parameter and model uncertainty. In contrast, described in Section 2.1.2, the Bayesian approach quantifies both aleatory and *epistemic uncertainties* by treating parameter estimates as random variables, thus providing a more comprehensive uncertainty estimation.

2.1.1.2 Quantifying Uncertainty for Deterministic Predictions

The probabilistic formulation for the linear case above is limited in its ability to capture the full range of uncertainty. A more comprehensive measure is the *prediction interval.* Given a new observation vector $\boldsymbol{x}_h$, the prediction interval for a new response, $\hat{y}_h$, quantifies the prediction's uncertainty. This interval is expressed as:

$$\hat{y}_h \pm t_{\alpha/2,n-2} \times \sqrt{\text{MSE} + [se(\hat{y}_h)]^2}$$

Here, $\hat{y}_h$ is the predicted value of the response for the feature values $\boldsymbol{x}_h$, $t_{\alpha/2,n-2}$ is the t-multiplier with $n-2$ degrees of freedom, where n is the sample size, and

MSE is the mean squared error from the regression. The term $\sqrt{\text{MSE} + [se(\hat{y}_h)]^2}$ can be expressed as μ_y, called the standard error of the prediction interval. This term incorporates both the standard error of the fit, $se(\hat{y}_h)$, and the observation noise, represented by the MSE. Thus, the equation expresses uncertainty in the mean prediction through $se(\hat{y}_h)$ and uncertainty due to observation noise through MSE.

The prediction interval can be used when $\boldsymbol{x}_h$ is within the scope of the model and when linearity, independence, normality, and equal variances of errors are satisfied. Notably, the condition of normally distributed error terms is crucial for the suitability of using the prediction interval.

The prediction interval for $\hat{y}_h$ at $\boldsymbol{x}_h$ tends to be wider than the *confidence interval*, which compared to the prediction interval has $\mu_y = se(\hat{y}_h)$. The larger width of the prediction interval, compared to confidence interval, results from the additional MSE term, reflecting the greater uncertainty when predicting an individual response rather than estimating a mean response. As the sample size n increases, the confidence interval may narrow, but the prediction interval cannot shrink to zero due to the inherent variability captured by the MSE.

2.1.2 Bayesian Approach

Bayesian methods are fundamentally probabilistic and facilitate decision-making under uncertainty by not only focusing on the most likely outcomes but also considering the distribution and likelihood of a wide range of possible outcomes. This approach is advantageous when data is sparse or when it is essential to integrate domain knowledge into the model.

The Bayesian approach treats parameters of interest, θ, as random variables with associated probability distributions that represent our knowledge or uncertainty about their values. This contrasts with the Frequentist perspective, which considers parameters as fixed but unknown quantities. The Bayesian paradigm incorporates prior beliefs about θ through a prior distribution, $p(\theta)$, which is updated to a posterior distribution, $p(\theta|D)$, given observed data D.

In supervised learning within the Bayesian framework, regression models start by positing a prior distribution over the parameters θ, which is then updated to the posterior distribution $p(\theta|D)$ using Bayes' theorem, reflecting the incorporation of data D. The full posterior predictive distribution can further be derived and expressed accordingly:

$$p(y|\boldsymbol{x}, D) = \int p(y|\boldsymbol{x}, \theta)p(\theta|D)d\theta, \tag{2.2}$$

where $p(y|\boldsymbol{x}, \theta)$ is the likelihood of the target variable y given $\boldsymbol{x}$ and θ, and $p(\theta|D)$ represents the posterior distribution of θ given the data. Integrating over all values of θ weighted by their posterior probability, yields the full predictive distribution, $p(y|\boldsymbol{x}, D)$.

The full posterior distribution, Equation 2.2, incorporates both epistemic uncertainty, by the uncertainty of the parameters θ, as well as the inherent noise in the data, aleatoric uncertainty. However, the full posterior distribution is frequently analytically intractable. Monte Carlo integration, variational inference or conjugate priors, if applicable, can be leveraged for approximation.

Another approach is Maximum A Posteriori (MAP) estimation, which only considers the single best guess for the parameters, θ_{MAP}, for inference. MAP estimation infers the predictive distribution seen in Equation 2.3. By design, MAP reduces the problem to the Frequentist paradigm since it assumes a deterministic parameter, thereby neglecting incorporation of the epistemic uncertainty.

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(\theta|D) \implies p(y|\boldsymbol{x}, \theta_{\text{MAP}}) \tag{2.3}$$

2.1.2.1 Hierarchical Linear Regression

Hierarchical Linear Regression (HLR) is an extension of Bayesian Linear Regression that excels at regression tasks where the data exhibit structured dependencies, by modeling these at multiple levels of hierarchy. For example, a HLR model can assign every grouping of data, j, unique parameter vectors, $\boldsymbol{\theta}_j$. Furthermore, along every dimension, the parameters $\theta_{j,d}$ share the same unknown hyperparameter, $\mathcal{E}_d$, with prior $p(\mathcal{E}_d)$ that capture the overall fixed effects across all groups, which is then updated to the posterior distribution, $p(\mathcal{E}_d|D)$, given the data [13].

The estimation of the model parameters $\boldsymbol{\theta}_j$ assumes that the effects at higher levels of the hierarchy can influence outcomes at lower levels. The variability of these parameters is typically modeled using variance components that define how much individual group estimates can deviate from the overall population effects. This hierarchical structuring of the regression model permits the explicit accounting of intra-group correlation and heterogeneity, often ignored in standard, pooled regression analyses.

A hierarchical model can compute the posterior predictive distribution, inferred from data D, and generate probabilistic predictions, $p(y_j|\boldsymbol{x}_j, D)$, for a group j:

$$p(y_j|\boldsymbol{x}_j, D) = \int p(y_j|\boldsymbol{x}_j, \boldsymbol{\theta}_j) p(\boldsymbol{\theta_j}|D) d\boldsymbol{\theta}_j \tag{2.4}$$

With information being shared between the parameters of different groupings, $\boldsymbol{\theta}_j$, via $\boldsymbol{\mathcal{E}}$, it is also possible to generate predictions for groups of data similar to existing groups, even if the model has not seen them during training. In such cases, the parameters of the new group n, $\boldsymbol{\theta}_n$, have to be inferred from the posterior distribution of $\boldsymbol{\mathcal{E}}$ [13]:

$$p(y_n|\boldsymbol{x}_n, D) = \int\int p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}_n) p(\boldsymbol{\theta}_n|\boldsymbol{\mathcal{E}}) p(\boldsymbol{\mathcal{E}}|D) d\boldsymbol{\theta}_n d\boldsymbol{\mathcal{E}} \tag{2.5}$$

2.1.2.2 Gaussian Process Regression

Gaussian Process Regression (GPR) utilizes a powerful Bayesian framework to model complex datasets through continuous stochastic processes. As a non-parametric kernel-based approach, GPR excels at providing a comprehensive quantification of uncertainty. Gaussian processes (GPs) serve as prior distributions that encapsulate beliefs about the form and structure of the underlying data-generating functions, which are updated to posterior distributions upon observing data [14]

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. This attribute enables GPR to extend multivariate Gaussian distributions to infinite-dimensional spaces, typically function spaces, making it ideal for regression in a high-dimensional setting.

Mathematically, a GP is defined by its mean function $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are points in the input space. The mean function provides a basic description of the expected value of the function at point $\boldsymbol{x}$, often assumed to be zero in the absence of prior information. The covariance function, or kernel, encodes assumptions about the function's properties, such as smoothness and the length scale over which function values are correlated:

$$m(\boldsymbol{x}) = \boldsymbol{E}[f(\boldsymbol{x})], \quad k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]$$

The Matérn kernel is a popular choice in Gaussian processes for its ability to handle multi-dimensional data and its control over the function's smoothness. It is defined by the following equation:

$$k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{r}{\ell}\right)^{\nu} K_{\nu}\left(\sqrt{2\nu}\frac{r}{\ell}\right)$$

where r is the distance between points, ℓ is the length scale parameter, ν is the smoothness parameter, and K_ν is a modified Bessel function. The parameter ν allows adjustment of the function's smoothness, which is critical for capturing different levels of detail in the data's underlying structure.

The training of a Gaussian Process model involves optimizing the hyperparameters of the kernel. This optimization is achieved by maximizing the log marginal likelihood (Equation 2.6) of the observed data, which balances the model's fit to the data with its complexity.

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top K^{-1}\boldsymbol{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{2.6}$$

where $\boldsymbol{y}$ are the observed targets (if many), $\boldsymbol{X}$ is the input feature data, K is the covariance matrix computed from the kernel function, and $\boldsymbol{\theta}$ represents the kernel hyperparameters. In Equation 2.6, neither $\boldsymbol{X}$ nor $\boldsymbol{\theta}$ explicitly appears on the right-hand side. The two parameters are implicitly represented through the covariance matrix K, which is a function of $\boldsymbol{X}$ and $\boldsymbol{\theta}$. The first term in the equation reflects the data fit, the second term penalizes model complexity, and the third term is a normalization constant.

Due to the non-convex nature of the log marginal likelihood, optimization can be challenging and often requires numerical techniques like gradient descent or quasi-Newton methods. These methods iteratively adjust the hyperparameters to find a local optimum of the likelihood function, crucial for the model's performance and accuracy in predictions.

2.1.2.3 Bayesian Neural Networks and Variational Inference

Bayesian Neural Networks (BNNs) leverage Bayesian inference to provide a probabilistic interpretation of neural network weights and biases, treating them not as fixed values but as distributions. This probabilistic framework enables the estimation of uncertainty in the network's predictions, differentiating it fundamentally from traditional neural networks. Hence, BNNs consider the distribution over all possible parameters, weighted by their posterior probabilities, to form a prediction as described mathematically in Equation 2.2.

Direct computation of the posterior distribution in BNNs is intractable due to the high dimensionality of the parameter space, a core characteristic of neural networks. Hence, approximation methods are employed such as variational inference and Markov Chain Monte Carlo (MCMC).

Variational inference is a technique used to approximate the posterior distribution of the weights, $\boldsymbol{\omega}$, with a simpler tractable distribution, rather than performing exact inference. Specifically, a variational distribution, $q(\boldsymbol{\omega})$, is defined to approximate the true posterior distribution $p(\boldsymbol{\omega}|D)$:

$$p(\boldsymbol{\omega}|D) \approx q(\boldsymbol{\omega}; \theta), \tag{2.7}$$

where θ denotes the parameters of the variational distribution, typically involving means and variances that are learned during training.

Commonly, the parameters of the variational distribution are optimzed to minimize the Kullback-Leibler divergence between $q(\omega; \theta)$ and $p(\omega)$, combined with with the expected log-likelihood of the data:

$$\mathcal{L}(\theta) = \mathbb{E}_{q(\omega;\theta)}[\log p(D|\boldsymbol{\omega})] - \mathrm{KL}(q(\boldsymbol{\omega}; \theta)\|p(\boldsymbol{\omega})). \tag{2.8}$$

This loss function, $\mathcal{L}(\theta)$, used in variational inference, combines data fit (measured by the likelihood term) and model complexity (regulated by the KL divergence term) to effectively balance the trade-off between them.

2.2 Evaluation of Probabilistic Calibration

Probabilistic calibration refers to the degree to which the predicted probabilities assigned by a model to events match the actual observed frequencies of those events. A model is considered perfectly probabilistically calibrated if the *probability integral transform* values are uniformly distributed [15]. This concept will be elaborated upon in section 2.2.1

In the following sections, techniques to evaluate probabilistic calibration are explained. These evaluation techniques are applicable to models that output one probability distribution per prediction, meaning each prediction has a unique probability density function (PDF). The evaluation techniques presented are of two types: qualitative and quantitative.

2.2.1 Histogram Analysis of CDF Values

This qualitative evaluation technique is designed for assessing models that output PDFs. It involves creating a histogram of "cumulative density function (CDF) values" to visualize how closely the models' predicted distributions match the true distribution. This approach is particularly effective for identifying the causes of poor calibration and understanding factors contributing to sub-optimal model performance. The histogram can highlight four potential issues: overestimated mean, underestimated mean, overestimated variance, and underestimated variance.

To create the histogram, for each prediction, the CDF of the PDF is derived. The CDF is then evaluated at the true label value corresponding to each prediction. These CDF values are then used to form the histogram. A well-calibrated model will produce a histogram that approximates a uniform distribution. If the histogram deviates from this uniform shape, it indicates that the model is not probabilistically calibrated [15]–[17].

The rational behind this evaluation method is grounded in the logic of the probability integral transform, which states that if a random variable is distributed according to the CDF of another random variable A it's distribution, B, is uniform on the unit interval [17]. In practical terms, this means that if you take different observations and apply the CDF of their respective predicted distributions, the resulting values should be uniformly distributed [16].

Let's denote the predicted distributions' CDFs by G_i for all i in the evaluation set and introduce the random variable Y_i for the observed true label value, which is distributed according to the true distribution. Further the CDF of the true distribution is denoted F. By the probability integral transform, the random variable $U \sim F(Y)$ is uniform. If instead, G_i is used as a proxy for F, meaning $U_i \sim G_i(Y_i)$, the collection of U_i values will be uniform if and only if the predictive distributions accurately model the true target distribution.

2.2.2 Continuous Ranked Probability Score (CRPS)

Continuous Ranked Probability Score (CRPS) is a *proper scoring rule* used to evaluate the quality of probabilistic forecasts, particularly in the context of continuous outcomes, as in regression [15], [18]. As a proper scoring rule, CRPS reaches its minimum value only when the predicted distribution from a model perfectly matches the true probability distribution. CRPS measures the discrepancy between a predictive CDF and the actual outcome represented by the observed data values. The formula to calculate CRPS for one prediction, i, is:

$$\mathrm{crps}(F_i, y_i) = \int_{-\infty}^{\infty} (F_i(x) - I(x \geq y_i))^2 \, dx \tag{2.9}$$

F_i is the CDF of the predicted PDF for a sample i. Further, y_i is the true label of that sample. Equation 2.9 computes the integral of squared differences between the cumulative probabilities and an unit step function, $I(\cdot)$. For example, in the case where the predicted distribution has Gaussian form, the farther the mean of the predicted distribution is from the actual observation y_i, the larger the penalty will be, resulting in a high CRPS. As stated, Equation 2.9 is only for one prediction. Hence, in order to achieve a measure for a model evaluated on a full dataset of size N, the individual results must be averaged:

$$\mathrm{CRPS} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{crps}(F_i, y_i) \tag{2.10}$$

2.3 Distributional Shift

Distributional shifts can take many forms, affecting the input features or the target variable separately, or simultaneously, and can be caused by a variety of different reasons [19]. The most common categories of distributional shifts are *covariate shift*, prior probability shift, concept drift, domain shift, and source component shift. Understanding the type of distributional shift present in the data is paramount to mitigating its effects.

Covariate shift occurs when the distribution of the input features changes between the training and test datasets, while the conditional distribution of the target variable given the input features remains the same. Formally, if a model is trained on a source domain, $\mathcal{S}$ and evaluated on a target domain, $\mathcal{T}$, this is expressed as: $p_{\mathcal{S}}(X) \neq p_{\mathcal{T}}(X)$ but $p_{\mathcal{S}}(Y|X) = p_{\mathcal{T}}(Y|X)$. The following issues may arise:

- **Training Bias:** The model is optimized for the source domain's distribution, not the target's. This bias occurs because the training process is tailored to the distribution of the features in the source domain, leading to suboptimal performance when the feature distribution shifts in the target domain.
- **Incomplete Representation of the Feature Space:** The source domain may not adequately cover all regions of the feature space that are important

in the target domain. As a result, the model fails to learn the relationships in those areas, leading to poor performance on the target data where those feature regions are prevalent.

- **Generalization Error:** Even if the model fits the source data well, it may overfit to source-specific patterns. This overfitting means that the model captures noise specific to the source domain, which do not generalize to the target domain. Consequently, the model's performance degrades on the target data.

Domain adaptation involves adjusting models to perform well on a target domain that differs from the training domain. This is essential when there is a covariate shift between the source and target domains. The goal of domain adaptation is to make the model robust to differences in the data distributions by aligning the source domain closer to the target domain in a way that reduces the impact of these differences. This can be achieved through various strategies that modify the learning process, adjust the data, or change the feature representations to bridge the gap between the domains. By effectively adapting the model to the new domain, it is possible to mitigate the negative effects of distributional shifts, improve generalization, and ensure that the model maintains high performance and reliability in the target domain. This process is crucial for applications where models are deployed in environments different from where they were originally trained, ensuring that they remain accurate and effective despite changes in the data distribution.

3
Methodology

To fulfill the objectives outlined in the Aim (Section 1.2), an experiment is designed to assess the models in adapting from a source domain to a target domain. Four distinct Cases, characterized by univariate, bivariate or multivariate *covariate shift*, are constructed, each comprising two datasets: one representing the source domain and the other the target domain. In the experiment, varying numbers of adaptation samples from the target domain are provided to adapt the model, followed by evaluation using the remaining portion of samples from the target domain dataset. Through this methodology, the efficacy and efficiency of different models in adapting to the target domain is evaluated. A methodological overview is delineated below in Section 3.1.

3.1 Methodological Overview

Defining S^k and T^k precisely: The thesis problem, described in Section 1.1, is explored through 4 different Cases defined by a source and target domain dataset, S^k and T^k, of collected data from drives in each domain. The labeled source domain dataset for each Case k is assumed to be samples from a true source domain distribution, consequently:

$$S^k = \{(\boldsymbol{x}_i^s, y_i^s) \sim p_{\mathcal{S}}(\boldsymbol{x}, y)\}_{i=1}^{N_{\mathrm{S}}} \tag{3.1}$$

We denote the entire theoretical source domain space as $\mathcal{S} = \{(\boldsymbol{x}, y) \mid (\boldsymbol{x}, y) \sim p_{\mathcal{S}}(\boldsymbol{x}, y)\}$. Thus, S^k is assumed to be a finite subset of observed samples, of size N_{S}, used for our experiment. Each sample, i (`drive`), is defined by a feature vector, $\boldsymbol{x}_i$ (drive conditions), and a label, y_i (energy consumption). Subsequently,

$$T^k = \{(\boldsymbol{x}_i^t, y_i^t) \sim p_{\mathcal{T}}(\boldsymbol{x}, y)\}_{i=1}^{N_{\mathrm{T}}} \tag{3.2}$$

constitutes the set of labeled target domain data available, of size N_{T}, assumed to be samples from the true target domain distribution, $p_{\mathcal{T}}(\boldsymbol{x}, y)$. Where the entire theoretical target domain space is denoted, $\mathcal{T} = \{(\boldsymbol{x}, y) \mid (\boldsymbol{x}, y) \sim p_{\mathcal{T}}(\boldsymbol{x}, y)\}$. The Case specific source and target domain datasets are constructed in Section 3.2 given the conditions outlined in the Problem, Section 1.1.

Model and adaptation technique selection: A few machine learning models are selected, they are introduced in Section 3.4 along with each model's corresponding adaptation technique.

The form of the probabilistic predictions: The model predictions, $\hat{p}(y|\boldsymbol{x})$, are assumed to take the form of Gaussian distributions, an assumption discussed in Section 3.6.

Structuring the experiment: To simulate the realistic scenario central for this thesis, only a subset of the target domain data is available, $T_{\text{adapt}} \in T^k$ (*test-drives*), and the disjoint residual set of samples is left for evaluation, $T_{\text{eval}} = T^k \setminus T_{\text{adapt}}$. In short, the models are trained on the source domain dataset, S^k, and adapted on the adaptation set of samples from target domain dataset, T_{adapt}, and evaluated on the evaluation set of samples, T_{eval}. In Section 3.3 the experimental procedure is presented in detail.

Choice of evaluation techniques: Methods for evaluating the models' predictive distributions, employed at the evaluation stage of the procedure, are described in Section 3.5.

3.2 Data

In this report we use a dataset consisting of about 35 000 drives with electrical trucks collected from real-world settings during year 2023 and 2024. A `drive`, also commonly referred to as a sample throughout the report, is an array of collected data from a "journey" with an electric truck from an origin location to a destination. Specifically, a `drive` is a set of features, $\boldsymbol{x}$. The dimension of the feature array is six (seven for Case 3). The feature names are masked due to the confidentiality of the dataset. However, for contextual understanding, they are categorized into the following groups:

Route profile features:

- `route feature #1`
- `route feature #2`

Vehicle setup features:

- `vehicle feature #1`
- `vehicle feature #2`
- `vehicle feature #3`
- `vehicle feature #4`

Weather condition features:

- `weather feature #1`

The label of each `drive`, y, is the `energy consumed kwh per km` of that instance. The choice of modeling energy consumption per kilometer (kWh/km) instead of total energy consumption (kWh) was made to normalize the energy usage with respect to the distance traveled.

$$\texttt{drive} = (\boldsymbol{x}, y), \text{ where } |\boldsymbol{x}| = 6 \text{ (or 7)} \tag{3.3}$$

Four Cases, scenarios, are constructed for the experiment. Each Case is defined by a source domain dataset, S^k, and a target domain dataset, T^k, which together exhibit covariate shift. The selection of `drives` for each Case will be discussed further in Section 3.2.1-4. We choose to evaluate the models on four Cases in order to obtain a more comprehensive view of how the models perform when the data is subject to various types of covariate shift. For example, the Cases differ in how many of the features that are identically distributed, not shifted, between the source and target domain datasets.

3.2.1 Case 1 - Temperature: Univariate Shift

Characteristics:

- N_S = 5885
- N_T = 499
- Univariate covariate shift caused by continuous feature: `weather feature #1` (strong shift, disjoint)

Case 1 sets out to assess the models in adapting to a target domain with significantly colder temperatures. The Case is designed, samples have been chosen specifically, to reduce the covariate shift over the other features, i.e., the route profile and vehicle setup features. By designing the Case in this way, the results observed can be better isolated to the effect of the particular, disjoint, shift in the weather feature between the source and target domain datasets. A comparison between the source and target domain datasets' feature distributions is illustrated in Figure 3.1. Case 2 (Temperature: Multivariate Shift) is a very similar Case but where other features are shifted as well.

Source domain dataset: Characterized by `drives` for which the temperature is warmer, filtered by applying a lower bound of X degrees celsius:

$$\texttt{X} \leq \texttt{weather feature \#1}$$

Target domain dataset: In contrast, the target dataset is characterized by `drives` where the temperature is colder, filtered by applying an upper bound of X-10 degrees celsius:

$$\texttt{weather feature \#1} \leq \texttt{X-10}$$

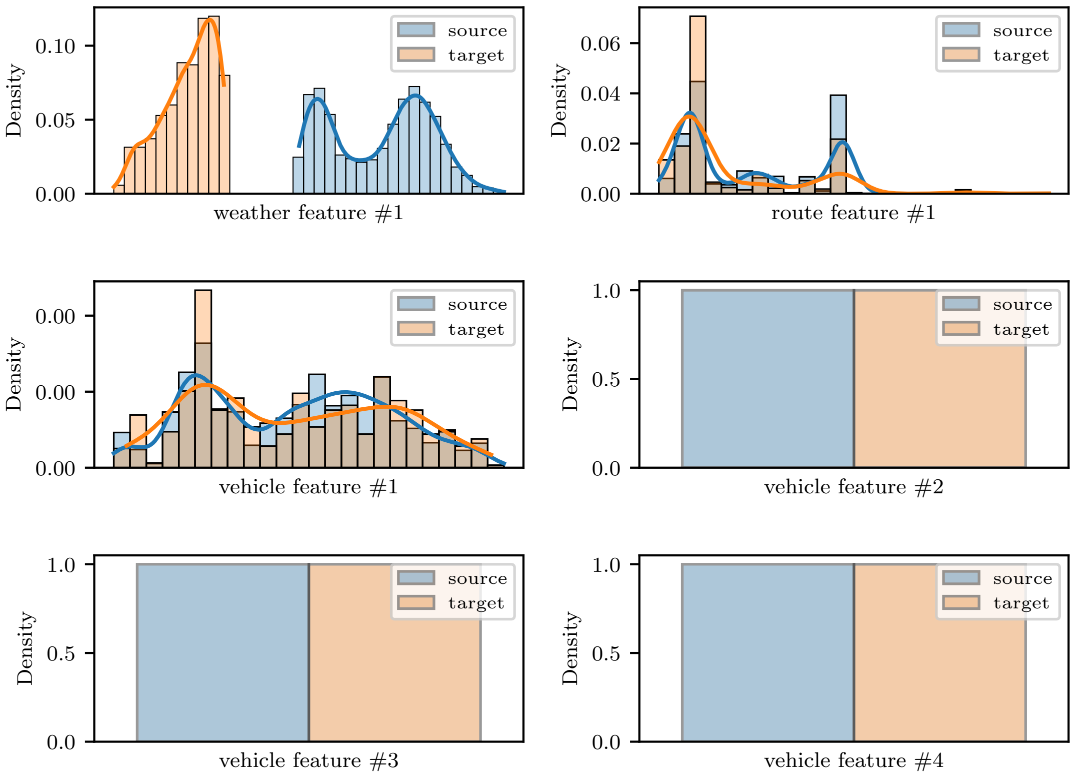


Figure 3.1: Feature distributions for all features in Case 1. `weather feature #1` is shifted.

3.2.2 Case 2 - Temperature: Multivariate Shift

Characteristics:

- N_S `= 6488`
- N_T `= 402`
- Multivariate covariate shift, primarily caused by continuous features: `weather feature #1` (strong shift, disjoint, continuous feature) and less significant shifts for other features

Similar to Case 1, this Case sets out to assess the models in adapting to a target domain with colder temperatures, but it is not designed to isolate the feature distribution shift to one feature. Namely, `drives` within the source and target domain datasets have been selected to be geographically separated, simulating the situation of expanding business operations by introducing a new route in a colder climate. Hence, no actions were taken to mitigate the covariate shift induced by this selection in the other features, resulting in a multivariate covariate shift rather than a univariate covariate shift as in Case 1.

Source domain dataset: Characterized by `drives` for which the temperature is

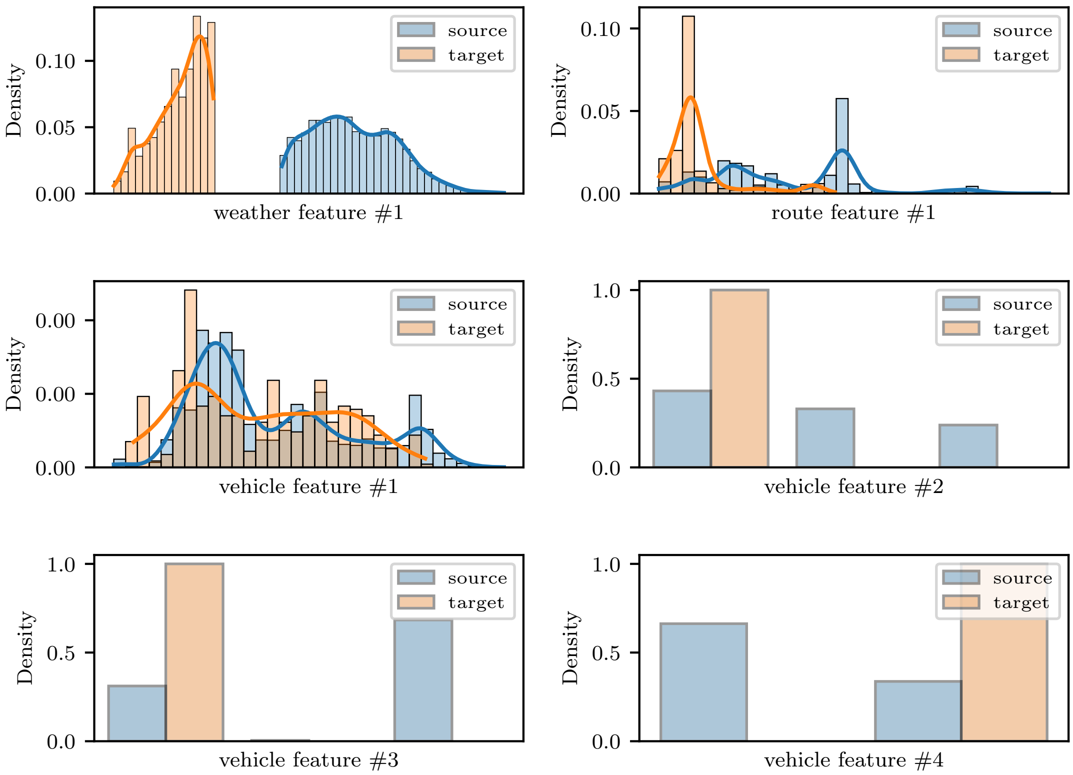


Figure 3.2: Feature distributions for all features in Case 2. All features are shifted to various degrees

warmer, filtered by applying a lower bound of X degrees celsius:

$$\texttt{X} \leq \texttt{weather feature \#1}$$

In addition, the `drives` are within a geographical area, qualitatively assessed, realistic for an electric truck fleet management company.

Target domain dataset: In contrast, the target dataset is characterized by `drives` where the temperature is colder, filtered by applying an upper bound of X-10 degrees celsius:

$$\texttt{weather feature \#1} \leq \texttt{X-10}$$

Further, the `drives` within the target domain dataset are geographically disjoint from `drives` within the source domain dataset. Implying, that the `drives` from the two sets have not been collected on the same routes. Therefore, there is a significant feature distribution shift within the route profile feature group. Figure 3.2 illustrates the feature distribution between the source and target domain datasets.

3.2.3 Case 3 - New Type of Route

Characteristics:

- N_S = `4218`
- N_T = `4221`
- Bivariate covariate shift caused by continuous features: `route feature #1` (weak shift) and `route feature #2` (strong shift)

Case 3 sets out to assess the models in adapting to a target domain with a different route profile. Different types of driving, city versus highway for example, is known to impact the energy consumption of trucks to various extent, motivating the inclusion of this Case in the thesis. Similar to Case 1, this case has been designed to isolate the covariate shift to specific features. Moreover, the label distributions of the source and target domain datasets of this Case are closely aligned. For a comparison of the feature distributions of the source and target domain datasets, see Figure 3.3.

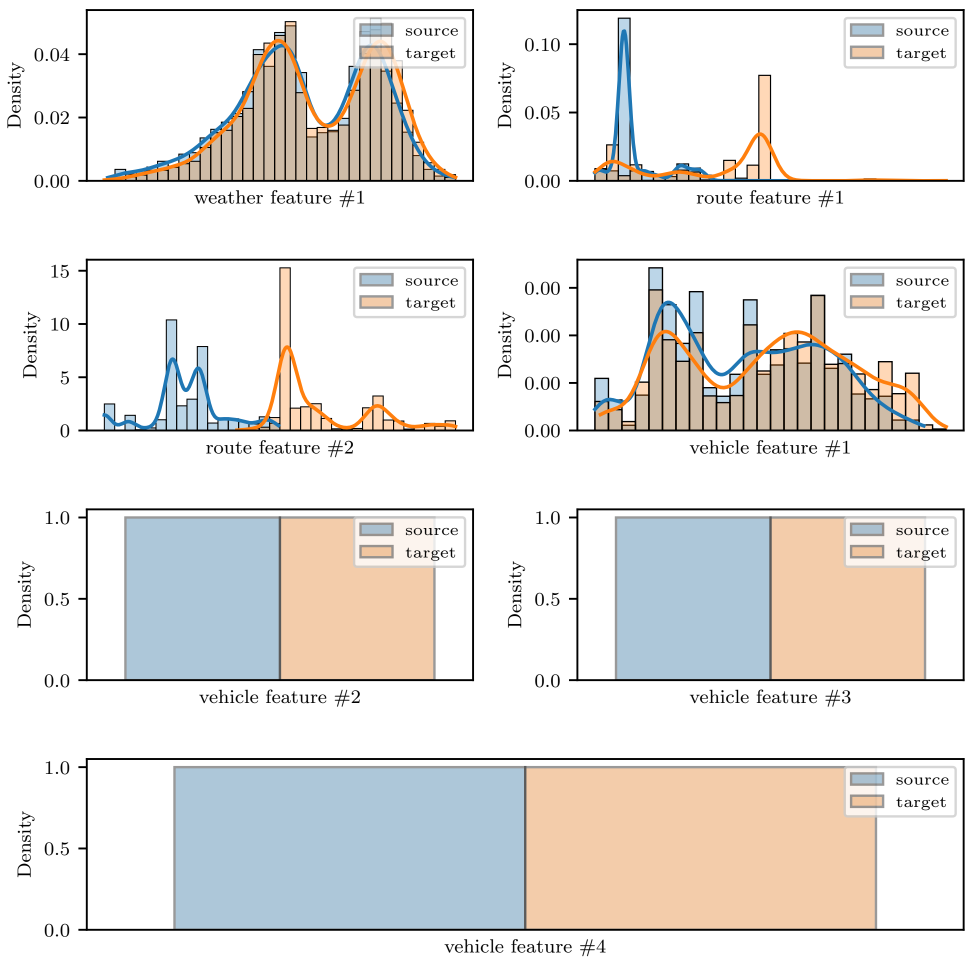


Figure 3.3: Feature distributions for all features in Case 3. `route feature #1` and `route feature #2` are shifted.

Source domain dataset: Characterized by `drives` for which the majority of the driving is done on regional or city roads.

Target domain dataset: In contrast, the target dataset is characterized by `drives` for which the majority of the driving is done on highway.

3.2.4 Case 4 - New Vehicle Manufacturer

Characteristics:

- N_S `= 11062`
- N_T `= 2197`
- Multivariate covariate shift, primarily caused by categorical feature: `vehicle feature #2` (strong shift, disjoint, categorical feature) and significant shifts for the other features

This Case sets out to assess the models in adapting to a target domain consisting of a different, unseen, vehicle manufacturer brand. The rationale behind this Case is that it is a plausible scenario, for an electric truck fleet management company, to want to integrate electric trucks from a completely new truck manufacturer into their fleet. Similar to Case 2, this Case is supposed to be more realistic, and thus, the Case has not been designed to isolate the covariate shift to a subset of features. In effect, this Case exhibits multivariate covariate shift. For a comparison of the feature distributions of the source and target domain datasets, see Figure 3.4. Notably, the `vehicle feature #2` is categorical and thus, one-hot encoded, when modelled on. Consequently, the feature column corresponding to new vehicle manufacturer of each `drive` in the source domain dataset is 0, whilst 1 in the target domain dataset.

Source domain dataset: Characterized by `drives` driven with electric trucks from all manufacturers but A:

$$\texttt{vehicle\_setup \#2} = \text{B or } \texttt{vehicle\_setup \#2} = \text{C}$$

Target domain dataset: In contrast, the target dataset is characterized by `drives` driven with electric trucks from manufacturer A:

$$\texttt{vehicle\_setup \#2} = A$$

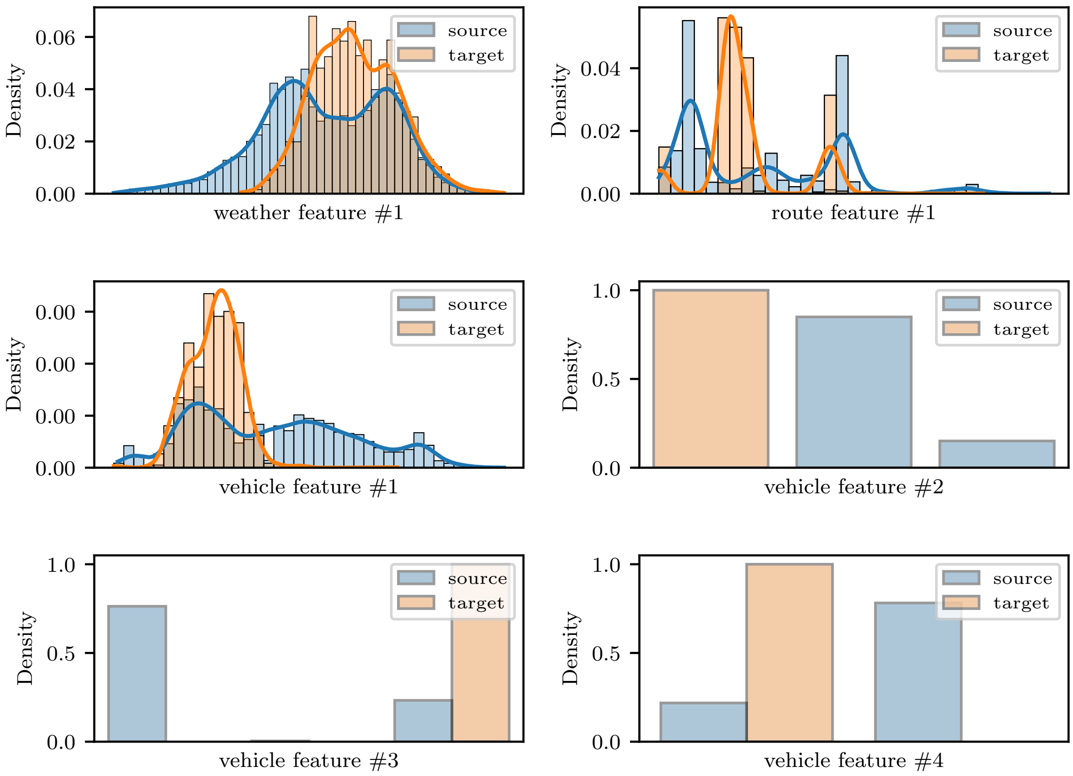


Figure 3.4: Feature distributions for all features in Case 4. All features are shifted to various degrees.

3.3 Experimental Procedure

The experiment sets out to evaluate the models on each Case k, consisting of S^k and T^k described in the prior section. The pseudo-code in Algorithm 1 describes the result generating process (per Case and model) and the text below delves into more detail. The model specific training and adaptation steps are described in Section 3.4).

The source domain dataset for Case k is used to train each model, $\mathcal{M}_0$.

$$\mathcal{M} = \text{Train}(\mathcal{M}_0, S^k)$$

A subset of the target domain dataset, T^k, is used for adaptation. In order to evaluate the models' sensitivty to the amount of data that is available for the target domain, we evaluate across a range of adaptation sample sizes, `n_adapt`, with 50 as the largest size. In total, 50 adaptation samples are sampled randomly from the target domain dataset resulting in, T_{adapt}. The samples left in the target domain dataset are used for evaluation, T_{eval}. The sampling of T_{adapt} is random.

$$
\begin{aligned}
&\texttt{n\_adapt} = \{0, 1, 3, 5, 10, 30, 50\} \\
&T_{\text{adapt}} \subset T^k \quad \text{where } |T_{\text{adapt}}| = 50 \\
&T_{\text{eval}} = T^k \setminus T_{\text{adapt}}
\end{aligned}
$$

The set of adaptation samples is used to adapt the model for various sizes of adaptation samples.

$$
\begin{aligned}
&T_{\text{adapt}}^{(n)} \subseteq T_{\text{adapt}}, \quad |T_{\text{adapt}}^{(n)}| = n \\
&T_{\text{adapt}}^{(0)} \subset T_{\text{adapt}}^{(1)} \subset T_{\text{adapt}}^{(3)} \subset \cdots \subset T_{\text{adapt}}^{(50)}
\end{aligned}
$$

We adapt the source trained model, $\mathcal{M}$, using the adaptation samples $T_{\text{adapt}}^{(n)}$:

$$\mathcal{M}^{(n)} = \text{Adapt}(\mathcal{M}, T_{\text{adapt}}^{(n)})$$

The adapted model, $\mathcal{M}^{(n)}$, is then evaluated on the evaluation set, T_{eval}, to obtain the quantitative performance metrics (Section 3.5.1) for the specific Case and model:

$$\mathcal{P}^{(n)} = \text{Evaluate}(\mathcal{M}^{(n)}, T_{\text{eval}})$$

In order to reduce bias from the selection of T_{adapt} (and in effect, T_{eval}) the experiment is run 30 times where T_{adapt} each time is sampled randomly without replacement (until all samples has been sampled once, then reset). Lastly, the performance metrics means and standard deviations over the 30 iterations, for each adaptation sample size, $\mathcal{P}^{(n)}$, are calculated.

3.4 Model Selection and Implementation

In the following sections, the models employed in the experiment are introduced. Beyond providing insights into the frameworks and packages used to implement them, special attention is given to motivating the choice of models and describing their specific adaptation techniques.

3.4.1 Linear Baselines

Two baseline models are utilized: the *source-trained linear baseline model* and the *target-only baseline model*. We employ linear Ridge regression models because of their simplicity and their common use as baselines. The source-trained baseline model is trained on the source dataset, S, while the target-only model is trained solely on the target adaptation samples, T_{adapt}. The target-only baseline is employed to explore the relevance of source domain training in the linear case. In addition, it is relevant how dependent such a domain specific model is on the number of samples from the target domain that are available. The target-only model provides a benchmark,

Algorithm 1: Experiment implementation, for specific Case and model

Input: Model $\mathcal{M}_0$, Source dataset S^k, Target dataset T^k
Output: Evaluation metrics

$(x_{\text{train}}, y_{\text{train}}) \leftarrow S^k$;
`model` $\leftarrow \mathcal{M}_0$;
`model_source_trained` = `model.train(`$x_{\text{train}}, y_{\text{train}}$`)`;
foreach *iteration i in range(30)* **do**
  $T_{\text{adapt}} \leftarrow$ sample batch of 50 `drives` from T^k;
  $T_{\text{eval}} \leftarrow T^k \setminus T_{\text{adapt}}$;
  $(x_{\text{adapt}}, y_{\text{adapt}}) \leftarrow T_{\text{adapt}}$;
  $(x_{\text{eval}}, y_{\text{eval}}) \leftarrow T_{\text{eval}}$;
  foreach n_{adapt} *in [0, 1, 3, 5, 10, 30, 50]* **do**
    `model` $\leftarrow$ `model_source_trained`;
    if $n > 0$ **then**
      `model.adapt(`$x_{\text{adapt}}[0 : n_{\text{adapt}}], y_{\text{adapt}}[0 : n_{\text{adapt}}]$`)`;
    $(\text{pred}_{\text{means}}, \text{pred}_{\text{stdevs}}) \leftarrow$ `model.predict(`x_{eval}`)`;
    $\text{metrics}[i][n_{\text{adapt}}] \leftarrow$ `calculate_metrics(`$y_{\text{eval}}, \text{pred}_{\text{means}}, \text{pred}_{\text{stdevs}}$`)`;
return `mean`(metrics), `std`(metrics);

applicable given the outlined delimitation of only considering model performance in the target domain.

Both models are implemented using Ridge regression from the `sklearn` library, chosen for its stability provided by the L2-norm regularization term, with the regularization parameter set to its default value, `alpha=1.0`. The loss function used for training these models is:

$$L(S, \theta) = \sum_{(x,y)\in S} (y - f(\boldsymbol{x}; \theta))^2 + \lambda \|\theta\|^2 \tag{3.4}$$

Ridge regression is deterministic and does not inherently produce probabilistic predictions. To enable comparison with other models, we estimate the uncertainty of deterministic predictions within the Frequentist framework. This involves employing the uncertainty estimation technique **prediction interval**, described in Section 2.1.1.2.

Adaptation technique: The adaptation to samples from the target distribution, T_{adapt}, is straightforward. The models are retrained on the combined dataset $S \cup T_{\text{adapt}}$ using the `fit()` method, minimizing the loss function described in Equation 3.4. Note that for the target-only baseline model, $S = \emptyset$. Mathematically, this adaptation process can be formulated as:

$$\theta_{\text{train}} \quad \rightarrow \quad \theta_{\text{adapted}} = \arg\min_{\theta} L(S \cup T_{\text{adapt}}, \theta) \tag{3.5}$$

3.4.2 Hierarchical Bayesian Linear Regression

Hierarchical models enhance estimation accuracy and efficiency by leveraging information across different groupings within the data. In scenarios where only a small amount of target data may be available, employing a hierarchical model holds promise for improving the domain adaptation ability of a linear model given hierarchical data structure. Case 4, which involves different vehicle types, exemplifies a situation where hierarchical modeling is especially advantageous, as it allows the model to utilize shared information across vehicle groups to improve predictions with limited target domain data.

The Hierarchical Bayesian Linear Regression (HLR) model in this thesis is implemented using the `PyMC` library, which is specifically designed for Bayesian modeling in Python. `PyMC` supports advanced Markov chain Monte Carlo (MCMC) and variational inference algorithms, making it suitable for the precise and efficient estimation of model parameters, despite the increased computational complexity with added dimensions in a hierarchical model.

The implemented HLR model is fundamentally a Bayesian linear regression model, distinguished by how the model parameters are distributed hierarchically. Some model parameters are modeled hierarchically, by vehicle manufacturer, meaning each type of vehicle manufacturer has its own set of model parameters. The decision to model specific dimensions hierarchically was informed by expert insights from Einride.

All HLR model parameters are assumed to follow a Gaussian distribution. This choice, the Gaussian distribution, is widely assumed because it is mathematically tractable and a natural choice for representing uncertainty and variability. For those modeled hierarchically, hyperpriors were set on the mean and standard deviations, reflecting the shared effects between different hierarchical levels (vehicle manufacturer). These hyperpriors were weak Gaussian priors, $\mathcal{N}(0,1)$.

The model was trained using the No-U-Turn Sampler (NUTS), an MCMC algorithm efficient for high-dimensional distributions and requiring no manual tuning [20]. NUTS was used to sample posterior distributions of the model parameters given the source domain data, which were then utilized to generate predictions. Unlike traditional linear regression implementations, where predictions are directly computed by inputting new data into a regression equation, the `PyMC` implementation samples from the posterior distributions of the model parameters for inference.

Adaptation technique: The HLR model was adapted to the target distribution, T_{adapt}, by using the posterior model parameter distributions, from training on the source domain, as new priors for adaptation. The same sampling procedure as during training was then run but with the adaptation samples.

To conform with `PyMC`'s framework, Kernel Density Estimation (KDE) was performed on the output from the previous training sampling round using the `gaussian_kde()` function from `scipy.stats`. Each prior affected by hierarchical levels was estimated per level and then concatenated to maintain the hierarchical structure.

3.4.3 Gaussian Process Regression

Gaussian Processes (GPs) constitute a family of non-parametric models that do not impose assumptions regarding the underlying relationships within the data, distinguishing them from parametric models such as linear models, which presume linear relationships. A Gaussian Process Regression (GPR) model was selected to investigate whether its capability to capture arbitrary relationships can be leveraged to perform well on a FSDA task. Additionally, GPs are *generative* models, excelling at accurately modeling the data on which they are trained. We hypothesize that this property enhances the model's adaptation sensitivity.

The GPR model in this study is implemented using the `GPyTorch` library, which is built on `PyTorch` and offers efficient training and inference through GPU acceleration. `GPyTorch`'s modular design allows for tailored model implementation to meet the thesis objectives. Simpler GPR implementations, such as those in `Scikit-learn`, were deemed impractical due to long training times and excessive memory usage. Given the size of the Case datasets experimented on it was possible to employ a GPR model using exact inference, `gpytorch.models.ExactGP`, in contrast to requiring approximation techniques.

A Matérn kernel was selected to work well with the multidimensional data, with the smoothness parameter, ν, set to 1.5 after tuning as this balanced underfitting and overfitting effectively. The likelihood of the GPR model was Gaussian, aligning with the Bayesian regression framework and enabling uncertainty estimates of predictions. The noise parameter of the likelihood, σ^2, was set to the default value of 0.6932.

The model was trained over 50 iterations using the Adam optimizer with an initial learning rate of 0.1. The loss function used was the `ExactMarginalLogLikelihood` (see Equation 2.6), provided by `GPyTorch`.

Adaptation technique: The adaptation technique of the GPR model utilizes the built-in method `get_fantasy_model(new_x, new_y)` from `GPyTorch`. This method for `gpytorch.models.ExactGP`, inspired by Online Variational Conditioning (OVC), allows the GPR model to incorporate new observations without retraining from scratch. OVC updates the GP's posterior distribution to include new data points by parameterizing the mean and covariance of a variational GP for additive updates of the posterior predictive distribution. Note that this method was originally proposed for sparse variational GPR (SVGPR), but the `GPyTorch` implementation allows its application to exact GP implementations as well, as exact GP can be considered a special case of SVGPR. For a more in-depth explanation of OVC, we refer to the paper by Maddox, Stanton, and Wilson [21].

3.4.4 Bayesian Neural Network

A Bayesian Neural Network (BNN) is selected for its inherent probabilistic nature and complexity. Unlike the generative Gaussian Process Regression (GPR) model, BNNs are *discriminative*, directly modeling the conditional probability $p(y|x)$. This capability allows BNNs to effectively learn the essential dynamics of the task, potentially outperforming simpler models by balancing detailed data representation with the inherent ability of probabilistic modeling. For additional details about BNNs, see Section 2.1.2.3.

The BNN in this thesis is implemented using `TensorFlow` and `TensorFlow Probability`, leveraging advanced functionalities tailored for probabilistic deep learning. The model architecture includes `DenseVariational` layers (`units = [8,8]`), employing variational inference to approximate the posterior distributions of the network weights. These layers capture *epistemic uncertainty* (uncertainty in the model parameters) effectively. Additionally, by modeling the output as an `IndependentNormal`, the model incorporates *aleatoric uncertainty* (inherent variability in the data), providing comprehensive uncertainty estimates in the predictions.

The model is compiled for training with a loss function that combines the `negative log-likelihood` of the predictions with a KL divergence term, ensuring the balance between the model fitting the data well and maintaining a regularized complexity. The `batch_size` is set to `32` and `learning_rate=0.01`. Regarding `num_epochs`, early-stopping technique is used to tune the parameters and avoid overfitting, using the following criteria: `monitor='val_loss'`, `min_delta=0.001`, `patience=15`, `verbose=1`, `mode='min'`, `restore_best_weights=True`.

Adaptation technique: The BNN model incorporates a custom loss function with a regularization component aligning the adapted model parameters, θ_{adapted}, with those obtained from the source data, θ_{train}. This regularization technique, developed and introduced by Lucas, Pelletier, Schmidt, *et al.* [22], is tailored to our experiment by adjusting the hyperparameter controlling the regularization strength, λ. This hyperparameter counteracts significant divergence from the source-trained parameters during adaptation to the target domain. The loss function of the regularization technique is defined as follows:

$$\text{SourceRegLoss}(x, y, \theta_{\text{adapted}}, f_\theta, \theta_{\text{train}}, \lambda) = L(f_\theta(x), y) + \lambda \|\theta_{\text{adapted}} - \theta_{\text{train}}\|^2 \quad (1)$$

where f_θ represents the adapted model and θ_{adapted} its parameters. During adaptation, the hyperparameters are set as follows: `λ = 10` (instead of none), `batch_size = 1` (instead of 32), and `num_epochs = "adaptive"` (instead of "early stopping"). The batch size is set to 1 to accommodate the small T_{adapt}, and the number of epochs is set adaptively, to ensure an equal number of gradient updates, `grad_updates = 5000`, across different sizes of adaptation samples, $T_{\text{adapt}}^{(n)}$ [22].

3.5 Evaluation Methods

Point prediction models can be assessed using metrics like R^2-score and mean absolute percentage error (MAPE). While these metrics can be applied to probabilistic models by treating the distributional expectation as the point prediction, they fail to capture how well models estimate prediction uncertainty. Consequently, these metrics do not provide insights into probabilistic calibration. Therefore, additional techniques are necessary for evaluating probabilistic predictions. To comprehensively understand model performance, both probabilistic and non-probabilistic evaluation methods are employed, as well as a combination of quantitative and qualitative evaluation techniques.

3.5.1 Quantitative Evaluation

Four quantitative evaluation metrics are used to evaluate the models' predictions: Wasserstein score, continuous ranked probability score (CRPS), MAPE and R^2-score. The Wasserstein score and CRPS specifically evaluate the probabilistic calibration of the models, while MAPE and R^2 assess predictive accuracy. Together, the metrics provide a holistic view of the models' predictive robustness, allowing detailed assessment of the models selected. Moreover, quantitative evaluation techniques are suitable to effectively rank and compare the models.

To quantify how well the histogram of CDF values (Section 2.2.1) conform to the uniform distribution, we employ a modification of the Wasserstein distance, a metric that measures the disparity between two distributions. However, since the Wasserstein distance between any given distribution and the uniform distribution on the unit interval is capped at 0.5, adjustments are required to derive a metric ranging from 0 to 1. Consequently, we adopt the modification proposed by Wick, Kerzel, Hahn, *et al.* [16]:

$$\text{Wasserstein score} = 1 - 2 * \text{Wasserstein distance} \tag{3.6}$$

This modification facilitates the computation of an accuracy metric, which measure the level of alignment between the predictive distributions and the true distribution. The modified Wasserstein distance is referred to as Wasserstein score in the thesis. Further, a detailed explanation of CRPS is provided in Sections 2.2.2. MAPE and R^2 scores are derived by comparing the expectation of the predicted Gaussian probability density functions (PDFs) with the true label values.

3.5.2 Qualitative Evaluation

Three qualitative evaluation techniques were employed to assess model performance: the histogram of Cumulative Distribution Function (CDF) values, the prediction interval plot and the predictive distribution plot. These techniques assess probabilistic calibration by offering detailed analysis of model predictions. They are crucial for identifying model tendencies, such as under- or overpredicted means and standard deviations of the predicted Gaussian distributions.

Histogram of CDF values: This technique visually represents the distribution of CDF values obtained by evaluating the CDFs of predicted Gaussian PDFs at the true label value. A uniform histogram of CDF values implies predictions that are probabilistically well calibrated. A more detailed explanation is found in Section 2.2.1, and four examples of this evaluation technique is presented in Figure 3.5. The histogram of CDF observations provides high-level insights into model calibration, enhancing the understanding of the predictions probabilistic characteristics.

Prediction interval plot: This technique helps illustrate the predictions intervals and assess the level of probabilistic calibration visually. Implemented as a scatter plot, this plot facilitates comparison between predicted and actual values by plotting the label values according to the y-axis and arranging the x-axis for readability without encoding specific values. The y-axis values are masked. Data points are sorted based on predicted label values, with two points plotted at the same x coordinate for each sample — one for the true label value and another for the predicted mean. The plot also includes a 95% prediction interval derived from the predicted standard deviations, providing insight into the prediction uncertainty of the predictions. Figure 3.6 visualizes an example of this plot.

Predictive distribution plot: This evaluation method is utilized to analyze how individual model predictions change with different numbers of adaptation samples. Figure 3.7 shows an example of this evaluation plot: for the same `drive`, the predicted Gaussian distributions from each adaptation run is displayed, illustrating changes in the mean and standard deviation of models' predictions. Also included in the plot is a vertical red line indicating the true label value of the `drive`. The plot allows assessment of model sensitivity to adaptation and how adaptation impacts the accuracy of individual predictions.

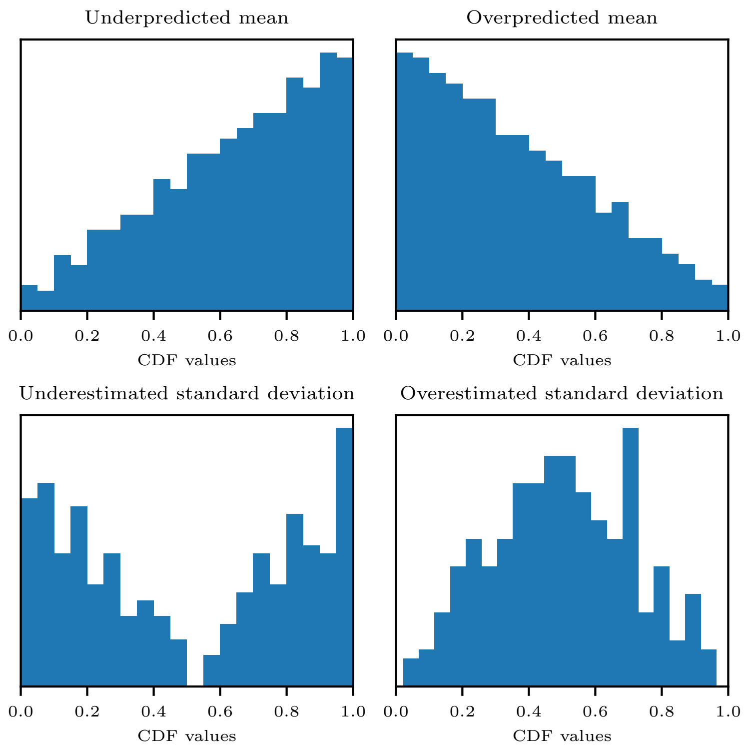


Figure 3.5: Example of histogram of CDF values plot, and the four cases of model deficiencies that can be identified with this evaluation technique.

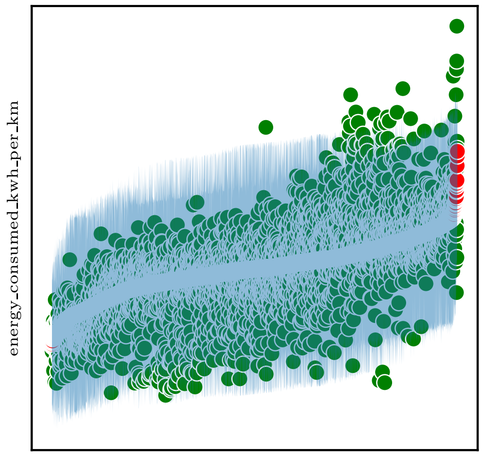


Figure 3.6: Example of prediction interval plot. Green dots indicate true label values, red dots show predicted label value and blue band indicates the 95% prediction interval.

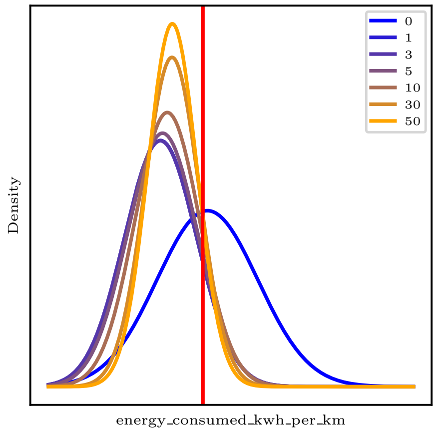


Figure 3.7: Example of plot illustrating how predictions (Gaussian distributions) for a `drive` changes across runs with increasing number of adaptation samples (blue to orange).

3.6 Gaussian Assumption for Model Uncertainty

The primary assumption for this study is that the uncertainty in the probabilistic estimates generated by the models follows a Gaussian distribution. This means that for each prediction, the models produce a mean value and an associated standard deviation, which together define a Gaussian distribution. This assumption ensures a consistent approach to representing uncertainty across different models and scenarios, simplifying the evaluation process and enhancing the comparability of results.

However, several critical considerations must be acknowledged regarding this assumption. Firstly, the assumption of Gaussian uncertainty may not always hold true in real-world scenarios where the underlying data distributions can exhibit skewness, kurtosis, or multi-modality. Relying on Gaussian distributions could potentially oversimplify the representation of uncertainty, leading to biased or inaccurate predictions in such cases [23]. Secondly, this assumption may limit the models' flexibility in capturing complex data patterns, as the Gaussian distribution is symmetric and cannot account for asymmetric uncertainties.

4

Results

In the following sections, the results from running the experiment outlined in Section 3.3 will be presented, Case-by-Case. In order to make the summarizing tables of metrics more readable, model names have been abbreviated. See Table 4.1 for a dictionary of model name abbreviations.

Abbreviation	Model name
BNN	Bayesian Neural Network
GPR	Gaussian Process Regression
HLR	Hierarchical Bayesian Linear Regression
ST	Source-Trained Linear Baseline
TO	Target-Only Linear Baseline

Table 4.1: Mapping of abbreviations used for model names in the Results chapter.

4.1 Case 1 - Temperature: Univariate Shift

In the following sections, we will present the results from conducting the experiment on Case 1. This Case is characterized by univariate *covariate shift*, caused by shift in the weather feature. Details are available in Section 3.2.1. Table 4.2 presents the performance metrics from the conducted experiment on Case 1.

4.1.1 Baseline Models

The *source-trained linear baseline model* performs, relatively, well on the task from start. Prior to introduction of adaptation samples from the target dataset, the model achieves MAPE of about 0.11 and is well calibrated probabilistically, indicated by the high Wasserstein score of 0.96. The introduction of samples from the target dataset, with lower temperatures exemplified by this Case, and adapting the model accordingly, does not alter the predictions of the model. Consequently, the metrics remain quite constant as seen in Table 4.2.

In contrast, the *target-only baseline model* have a steep learning curve where the metrics improve gradually. This is expected given that the model has not seen any data except for the target samples introduced in each experiment. When introduced to 50 samples from the target domain dataset, the model performs similarly to the

Metric	Model	Adaptation Samples						
		0	1	3	5	10	30	50
CRPS	ST	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05
	TO	n.a.	n.a.	n.a.	0.43 ± 0.42	0.20 ± 0.08	0.14 ± 0.07	0.13 ± 0.06
	HLR	0.15 ± 0.05	0.15 ± 0.04	0.15 ± 0.05	0.15 ± 0.04	0.15 ± 0.05	0.15 ± 0.04	0.15 ± 0.04
	GPR	0.20 ± 0.14	0.18 ± 0.18	0.18 ± 0.16	0.19 ± 0.13	0.20 ± 0.09	0.15 ± 0.06	0.14 ± 0.05
	BNN	0.13 ± 0.14	0.25 ± 0.18	0.12 ± 0.16	0.11 ± 0.11	0.10 ± 0.10	0.10 ± 0.08	0.07 ± 0.04
MAPE	ST	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00
	TO	n.a.	n.a.	n.a.	0.23 ± 0.11	0.15 ± 0.03	0.12 ± 0.01	0.12 ± 0.01
	HLR	0.11 ± 0.00	0.11 ± 0.01	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.01	0.11 ± 0.02	0.11 ± 0.02
	GPR	0.17 ± 0.14	0.16 ± 0.18	0.15 ± 0.16	0.14 ± 0.13	0.13 ± 0.09	0.11 ± 0.06	0.11 ± 0.04
	BNN	0.16 ± 0.14	0.16 ± 0.18	0.16 ± 0.16	0.16 ± 0.11	0.16 ± 0.10	0.16 ± 0.08	0.16 ± 0.04
R^2	ST	0.59 ± 0.01	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.01
	TO	n.a.	n.a.	n.a.	0.16 ± 0.33	0.36 ± 0.12	0.54 ± 0.03	0.58 ± 0.02
	HLR	0.59 ± 0.01	0.59 ± 0.01	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.01	0.59 ± 0.02	0.59 ± 0.02
	GPR	neg.	0.06 ± 0.25	0.24 ± 0.16	0.31 ± 0.13	0.39 ± 0.09	0.55 ± 0.06	0.60 ± 0.04
	BNN	0.16 ± 0.14	0.26 ± 0.18	0.26 ± 0.16	0.24 ± 0.11	0.27 ± 0.10	0.26 ± 0.08	0.24 ± 0.04
Wass	ST	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.01	0.96 ± 0.01	0.96 ± 0.01	0.95 ± 0.01	0.95 ± 0.01
	TO	n.a.	n.a.	n.a.	0.44 ± 0.06	0.83 ± 0.04	0.89 ± 0.05	0.89 ± 0.04
	HLR	0.95 ± 0.00	0.96 ± 0.01	0.95 ± 0.00	0.95 ± 0.00	0.95 ± 0.01	0.93 ± 0.02	0.94 ± 0.02
	GPR	0.57 ± 0.01	0.77 ± 0.18	0.75 ± 0.16	0.78 ± 0.13	0.78 ± 0.09	0.87 ± 0.06	0.91 ± 0.04
	BNN	0.56 ± 0.01	0.67 ± 0.25	0.73 ± 0.16	0.77 ± 0.11	0.85 ± 0.10	0.83 ± 0.08	0.91 ± 0.04

Table 4.2: Mean ± standard deviation of Wasserstein score, CRPS, MAPE and R^2-score across 30 iterations on the target evaluation dataset of Case 1, for each model adapted on various number of adaptation samples. For each size of adaptation samples and metric, the best score is highlighted.

source-trained baseline model in terms of mean accuracy. However, the target-only baseline model is worse calibrated as to estimate the prediction uncertainty indicated by a lower Wasserstein score and higher CRPS.

4.1.2 Hierarchical Bayesian Linear Regression Model

Similar to the source-trained baseline model, the HLR model attains high mean accuracy and is probablistically calibrated without being introduced to samples from the target distribution. The improvement by introducing adaptation samples is minimal, which in conjunction with the overall good performance indicates that the model has learned an accurate representation of how temperature affects the target variable. Figure 4.2 illustrates the inferred relationship between temperature and energy consumption for the different models.

4.1.3 Gaussian Process Regression Model

The GPR model performs poorly before adapted on samples from the target dataset. The predictions at this stage are characterized by being underestimations of the energy consumption (see Figure 4.1). Introduction of adaptation samples improves the model drastically, reaching similar MAPE as the source-trained baseline model when adapted on 50 adaptation samples. In addition, by adaptation, the model becomes more probabilistically calibrated as seen by the histograms in Figure 4.1.

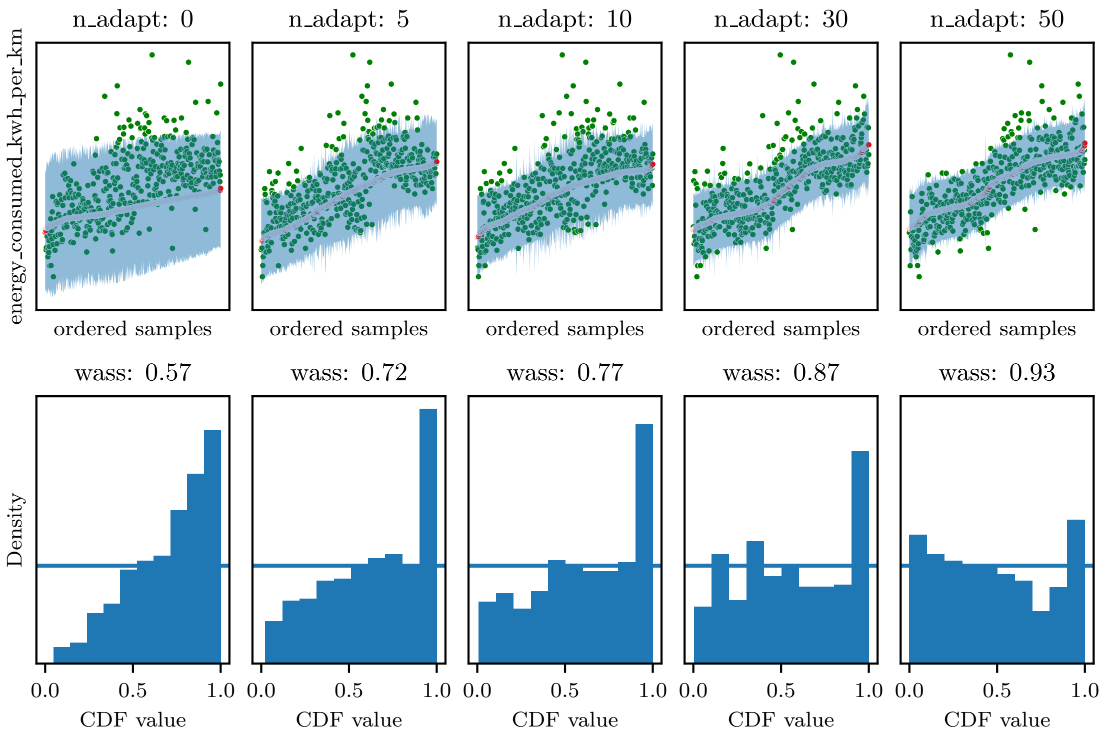


Figure 4.1: **Top row**: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 1 by the GPR model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty. **Bottom row**: Histograms of CDF values for the generated predictions.

Further, Figure 4.2 illustrates the inferred associations between temperature and energy consumption before and after adaptation. Before adaptation, the GPR model is clearly distinguished among the models by learning a positive association between temperature and energy consumption for colder temperatures. However, by adaptation, the model infers a target-feature dependency aligned with the other models.

4.1.4 Bayesian Neural Network Model

The BNN model, through adaptation, enhances the estimation of prediction uncertainty significantly more than the model improves its predictive accuracy. A small amount of the variability is explained by the model, low R^2 score, and the error in accuracy is significantly higher than for the other models. However, the probabilistic metrics, Wasserstein and CRPS, indicate a strong ability to estimate the uncertainty of its prediction.

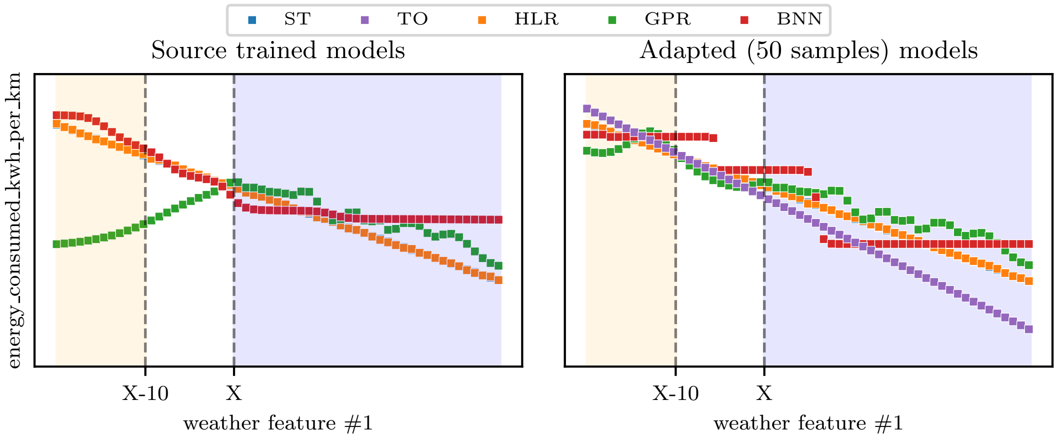


Figure 4.2: Predictions on simulated data, varying only the temperature while keeping all other features constant at their average values. The models are trained on Case 1's source dataset. The source and target domain temperatures are marked by the blue and yellow area respectively. **Left graph:** Un-adapted/only source-trained models (i.e. the TO model is not included). **Right graph:** The models adapted on 50 samples from Case 1's target dataset.

4.2 Case 2 - Temperature: Multivariate Shift

In the following sections, we will present the results from conducting the experiment on Case 2. This Case is characterized by multivariate covariate shift, primarily caused by shift in the weather feature. Details are available in Section 3.2.2. Table 4.3 presents the performance metrics from the conducted experiment on Case 2.

4.2.1 Baseline Models

Compared to the results attained by the baselines for the similar Case, Case 1, in Section 4.1.1, the source-trained baseline model does not demonstrate the same ability to accurately predict the energy consumption nor the uncertainty of it. In the case where 30 or 50 adaptation samples from the target domain dataset are introduced, the target-only baseline outperforms the source-trained model across most metrics. The source-trained baseline model displays low ability to adapt to the target dataset whereas the target-only model, not biased by the source dataset, showcase stronger performance with increasing amount of adaptation samples.

4.2.2 Hierarchical Bayesian Linear Regression Model

Relative to Case 1 (Temperature: Univariate Shift) in the previous section, the HLR model showcase comparable tendencies. The performance is relatively worse for this Case, but in line with what is observed for the source-trained baseline model.

Metric	Model	Adaptation Samples						
		0	1	3	5	10	30	50
CRPS	ST	0.20 ± 0.00	0.20 ± 0.00	0.20 ± 0.0	0.20 ± 0.0	0.19 ± 0.00	0.19 ± 0.0	0.18 ± 0.01
	TO	n.a.	n.a.	n.a.	0.19 ± 0.06	0.21 ± 0.07	0.08 ± 0.02	0.08 ± 0.02
	HLR	0.18 ± 0.0	0.16 ± 0.01	0.15 ± 0.02	0.15 ± 0.02	0.13 ± 0.04	0.12 ± 0.03	0.12 ± 0.03
	GPR	0.32 ± 0.0	0.14 ± 0.1	0.08 ± 0.06	0.09 ± 0.04	0.09 ± 0.04	0.07 ± 0.04	0.04 ± 0.02
	BNN	0.25 ± 0.0	0.07 ± 0.06	0.14 ± 0.09	0.14 ± 0.07	0.13 ± 0.04	0.10 ± 0.02	0.12 ± 0.02
MAPE	ST	0.13 ± 0.0	0.13 ± 0.0	0.13 ± 0.0	0.13 ± 0.0	0.13 ± 0.0	0.13 ± 0.0	0.13 ± 0.0
	TO	n.a.	n.a.	n.a.	0.17 ± 0.03	0.14 ± 0.03	0.12 ± 0.01	0.12 ± 0.01
	HLR	0.14 ± 0.0	0.14 ± 0.0	0.15 ± 0.0	0.15 ± 0.0	0.15 ± 0.01	0.14 ± 0.01	0.14 ± 0.01
	GPR	0.20 ± 0.0	0.15 ± 0.01	0.14 ± 0.02	0.14 ± 0.01	0.13 ± 0.02	0.12 ± 0.01	0.12 ± 0.01
	BNN	0.17 ± 0.0	0.23 ± 0.07	0.18 ± 0.02	0.16 ± 0.01	0.15 ± 0.01	0.14 ± 0.0	0.14 ± 0.01
R^2	ST	0.45 ± 0.02	0.46 ± 0.01	0.46 ± 0.01	0.46 ± 0.02	0.46 ± 0.02	0.47 ± 0.02	0.48 ± 0.02
	TO	n.a.	n.a.	n.a.	0.34 ± 0.14	0.48 ± 0.13	0.57 ± 0.05	0.59 ± 0.04
	HLR	0.43 ± 0.01	0.44 ± 0.01	0.44 ± 0.01	0.44 ± 0.01	0.44 ± 0.02	0.47 ± 0.02	0.48 ± 0.01
	GPR	neg.	0.3 ± 0.25	0.43 ± 0.19	0.45 ± 0.13	0.53 ± 0.07	0.59 ± 0.05	0.62 ± 0.02
	BNN	0.22 ± 0.01	neg.	0.24 ± 0.12	0.39 ± 0.06	0.47 ± 0.05	0.50 ± 0.03	0.49 ± 0.04
Wass	ST	0.72 ± 0.01	0.72 ± 0.01	0.73 ± 0.01	0.73 ± 0.02	0.73 ± 0.02	0.75 ± 0.02	0.76 ± 0.03
	TO	n.a.	n.a.	n.a.	0.45 ± 0.06	0.63 ± 0.14	0.85 ± 0.06	0.88 ± 0.08
	HLR	0.76 ± 0.01	0.77 ± 0.01	0.77 ± 0.01	0.77 ± 0.01	0.77 ± 0.02	0.78 ± 0.02	0.79 ± 0.01
	GPR	0.31 ± 0.02	0.63 ± 0.1	0.78 ± 0.06	0.73 ± 0.09	0.75 ± 0.12	0.76 ± 0.07	0.78 ± 0.05
	BNN	0.80 ± 0.01	0.55 ± 0.23	0.72 ± 0.1	0.79 ± 0.08	0.87 ± 0.11	0.90 ± 0.01	0.86 ± 0.07

Table 4.3: Mean ± standard deviation of Wasserstein score, CRPS, MAPE and R^2-score across 30 iterations on the target evaluation dataset of Case 2, for each model adapted on various number of adaptation samples. For each size of adaptation samples and metric, the best score is highlighted.

4.2.3 Gaussian Process Regression Model

Similarly to Case 1 in the previous section, the GPR model showcase poor performance on the target dataset before adaptation. For all metrics, the improvement observed throughout the inclusion of more adaptation samples is evident. The GPR model performs best for all metrics except the Wasserstein score in the run with 50 adaptation samples. Notably, the standard deviation of the scores, and by that the performance variability, is reduced with increasing number of adaptation samples.

Figure 4.4 illustrates a selected few samples for which the predictions, predictive distributions, of the GPR model are visualised across run with various number of adaptation samples. Considering the left example, the label distribution of the target domain is well distinguished from the label distribution of the source domain and the GPR model's accuracy for this sample improves substantially by increasing number of adaptation samples. Conversely, for the target sample in the right graph, the model fails when the target label is well represented in the source dataset and the predictions then get worse for setups with more adaptation samples. In addition, the two examples showcases how the estimated uncertainty in the predictions, the width of the predictive distribution, is narrowed by adaptation on more adaptation samples.

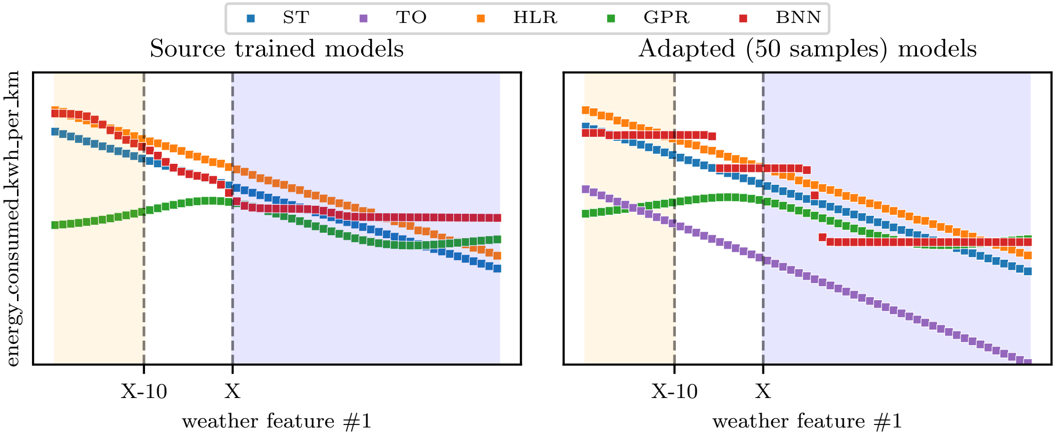


Figure 4.3: Predictions on simulated data, varying only the temperature while keeping all other features constant at their average values. The models are trained on Case 2's source dataset. The source and target domain temperatures are marked by the blue and yellow area respectively. **Left graph:** Un-adapted/only source-trained models (i.e. the TO model is not included). **Right graph:** The models adapted on 50 samples from Case 2's target dataset.

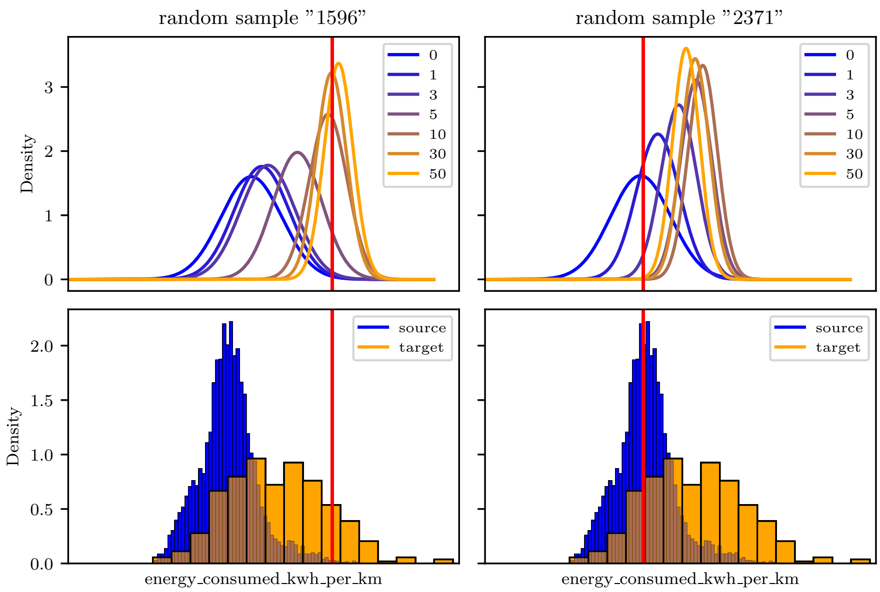


Figure 4.4: **Upper graphs:** Case 2, the GPR model's predictions (Gaussian distributions) for two different `drives` in the target domain evaluation set, across runs with increasing number of adaptation samples (blue to orange). **Lower graphs:** The label distribution of the source (blue) and target domain (orange) respectively. **Vertical line** (red)**:** illustrates the true label.

4.2.4 Bayesian Neural Network Model

For most metrics there is no extraordinarily strong trend as a function of the number of adaptation samples. The scores are quite stochastic, showcasing strong sensitivity to the number of adaptation samples included. This variability is illustrated for the Wasserstein score in Figure 4.5. In most cases, better scores are attained for 30 adaptation samples than for 50 adaptation samples, which combined with the general sensitivity observed hint at some level of overfitting by the model.

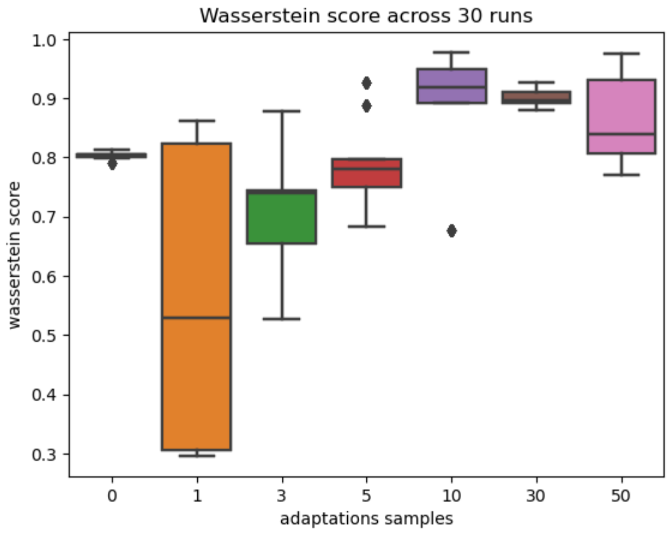


Figure 4.5: Boxplot of the Wasserstein scores for the BNN model on Case 2. The scores are those retrieved across the 30 runs for each configuration of adaptation sample size.

4.3 Case 3 - New Type of Route

In the following sections, we will present the results from conducting the experiment on Case 4. This Case is characterized by bivariate covariate shift, caused by shift in route features. Furthermore, the source and target domain label distributions are identically distributed. Details are available in Section 3.2.3. Table 4.4 presents the performance metrics from the conducted experiment on Case 3.

4.3.1 Baseline Models

Prior to adaptation, the performance of the source-trained baseline model is in line with the performance of other models. As in Case 1 and 2 however, the source-trained baseline model does not calibrate itself significantly after it gets adapted on an increased amount of adaptation samples. Nor does individual predictions change close to anything at all over the six adaptation rounds. However, slight improvements are made to model performance judging by the metrics.

The target-only baseline model achieves better final results than the source-trained baseline model and also among the best results of all models after 50 adaptation

Metric	Model	Adaptation Samples						
		0	1	3	5	10	30	50
CRPS	ST	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0
	TO	n.a.	n.a.	n.a.	0.12 ± 0.08	0.10 ± 0.04	0.07 ± 0.02	0.06 ± 0.01
	HLR	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.06 ± 0.01
	GPR	0.08 ± 0.0	0.07 ± 0.01	0.08 ± 0.02	0.08 ± 0.02	0.08 ± 0.02	0.06 ± 0.02	0.05 ± 0.01
	BNN	0.04 ± 0.0	0.25 ± 0.14	0.13 ± 0.11	0.11 ± 0.08	0.08 ± 0.04	0.07 ± 0.04	0.06 ± 0.02
MAPE	ST	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0
	TO	n.a.	n.a.	n.a.	0.19 ± 0.04	0.17 ± 0.06	0.12 ± 0.01	0.12 ± 0.01
	HLR	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.14 ± 0.0	0.13 ± 0.01	0.13 ± 0.0
	GPR	0.13 ± 0.0	0.13 ± 0.01	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.02	0.12 ± 0.01	0.12 ± 0.01
	BNN	0.14 ± 0.0	0.30 ± 0.06	0.27 ± 0.04	0.27 ± 0.03	0.26 ± 0.02	0.26 ± 0.01	0.26 ± 0.01
R^2	ST	0.46 ± 0.0	0.46 ± 0.0	0.46 ± 0.0	0.47 ± 0.0	0.47 ± 0.0	0.48 ± 0.01	0.50 ± 0.01
	TO	n.a.	n.a.	n.a.	neg.	0.07 ± 0.85	0.55 ± 0.08	0.57 ± 0.05
	HLR	0.46 ± 0.0	0.46 ± 0.01	0.47 ± 0.01	0.47 ± 0.01	0.48 ± 0.02	0.52 ± 0.03	0.56 ± 0.02
	GPR	0.48 ± 0.0	0.48 ± 0.02	0.40 ± 0.15	0.42 ± 0.13	0.43 ± 0.14	0.55 ± 0.05	0.59 ± 0.05
	BNN	0.41 ± 0.0	neg.	neg.	neg.	neg.	neg.	neg.
Wass	ST	0.87 ± 0.0	0.87 ± 0.0	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.88 ± 0.01	0.89 ± 0.02
	TO	n.a.	n.a.	n.a.	0.45 ± 0.06	0.79 ± 0.1	0.88 ± 0.06	0.93 ± 0.04
	HLR	0.87 ± 0.0	0.87 ± 0.01	0.87 ± 0.02	0.88 ± 0.03	0.88 ± 0.03	0.91 ± 0.04	0.93 ± 0.03
	GPR	0.82 ± 0.0	0.83 ± 0.01	0.83 ± 0.03	0.84 ± 0.04	0.86 ± 0.04	0.89 ± 0.03	0.91 ± 0.02
	BNN	0.88 ± 0.0	0.52 ± 0.18	0.74 ± 0.15	0.76 ± 0.11	0.85 ± 0.07	0.86 ± 0.06	0.89 ± 0.04

Table 4.4: Mean ± standard deviation of Wasserstein score, CRPS, MAPE and R^2-score across 30 iterations on the target evaluation dataset of Case 3, for each model adapted on various number of adaptation samples. For each size of adaptation samples and metric, the best score is highlighted.

samples on this Case. However, the performance of the target-only baseline model is worse when having seen only a small amount of samples from the target dataset, indicating that the other models benefits from having been trained on the source data.

4.3.2 Hierarchical Bayesian Linear Regression Model

For the hierarchical model there is a clear pattern of the model performing better on the target dataset as it gets adapted with an increasing number of adaptation samples from the target distribution. However, only adapting the model on few adaptation samples yields no significant calibration of the model, which shows in Table 4.4 as all metrics are stagnant between 0 to 10 adaptation samples. The large changes in model performance happens between 10 to 50 adaptation samples, especially between 30 to 50 adaptation samples. Furthermore, Figure 4.6 indicate that the hierarchical model underpredicts the distribution mean on drives in the target evaluation data when adapted on few adaption samples, and in later adaption runs, show signs of predicting too broad prediction uncertainties.

Although the model calibrates itself as it is introduced to more adaptation samples from the target dataset, individual predictions for the same target test data point does not change much. It is mainly the mean of the predicted distribution that

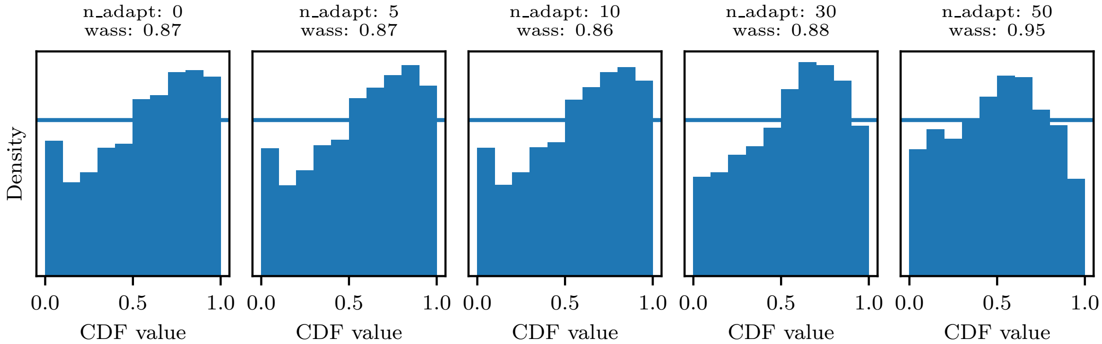


Figure 4.6: Histogram of CDF values derived from the HLR model's predictions in Case 3.

changes between the six different predictions in the same experiment iteration. The standard deviation on the other hand remains close to unchanged after adaptation, shown in Table 4.5. As Figure 4.6 indicates that the hierarchical model's predictions are too wide, i.e. large standard deviations, the improved calibration is explained by an improved ability to predict the Gaussian distribution mean.

# of adaptation samples	Average predicted standard deviation
0	0.223273
1	0.224679
3	0.225241
5	0.224996
10	0.227012
30	0.227549
50	0.224131

Table 4.5: Average standard deviation of the predictions (prediction uncertainty) per adaptation round, by the HLR model.

4.3.3 Gaussian Process Regression Model

Compared to other Cases, the Gaussian Process Regression (GPR) model achieves good probabilistic calibration on the target test data before any adaptation has happened, even though it is the least probabilistically calibrated out of all models. All metrics in Table 4.4 indicate that the model improves when being adapted on more adaptation samples, becoming better than the two baseline models according to all metrics but Wasserstein.

Although the Wasserstein and CRPS metrics from the evaluation after being adapted on zero adaptation samples are good, the predictions that the model produce are too broad which shows in the left-most sub-figures of Figure 4.7. On the one hand the prediction interval covers most of the target test data points, but on the other hand it does not model the data completely accurately. The right-most histogram in Figure 4.7, corresponding to to the predictions after the GP model has been adapted on 50 adaptation samples, indicate that the prediction uncertainties are still too broad at this point. However, examining the the prediction interval plots in the same figure, it is evident that the prediction interval shrinks drastically as the model is adapted, to the point where it appears that the model predicts overly narrow standard deviations, after 50 adaptation samples.

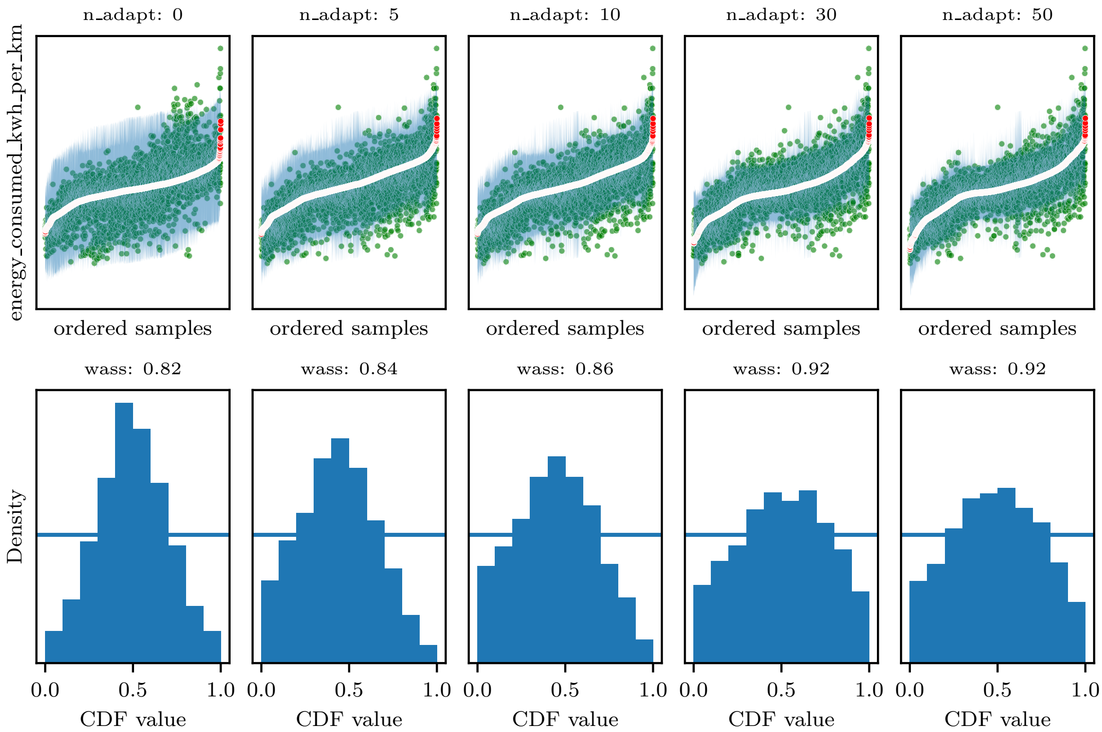


Figure 4.7: **Top row**: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 3 by the GPR model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty. **Bottom row**: Histogram of CDF values for the generated predictions.

4.3.4 Bayesian Neural Network Model

Looking at the metrics for the BNN model on Case 3 in Table 4.4, the model performs drastically worse on the target test data when adapted on one adaptation sample compared to only having been trained on the source data. Moreover, early in the adaption process, all metrics for the BNN model are worse than for other models. As the model gets adapted with more adaptation samples, the probabilistic calibration of the model becomes on par with that of the other models. However, the MAPE and R^2 scores do not improve much, being the worst of all models after 50 adaptation samples. Noteworthy is that the R^2-score is still negative at that point, indicating that the BNN model does not properly manage to explain the relationship between input features and label variable in the target data.

After adaptation, a notable discrepancy between the predicted value and the true label value exist for data points with a low true energy consumption, see Figure 4.8. The large predicted standard deviations for these data points suggests that the BNN model has high uncertainty about them. Further analyzes of these target domain data points show that they are short drives, similar to drives in the source data. However, such short drives in the source domain data have significantly higher energy consumption, which signals that the causation learnt during training on source data is not of relevance in the target data.

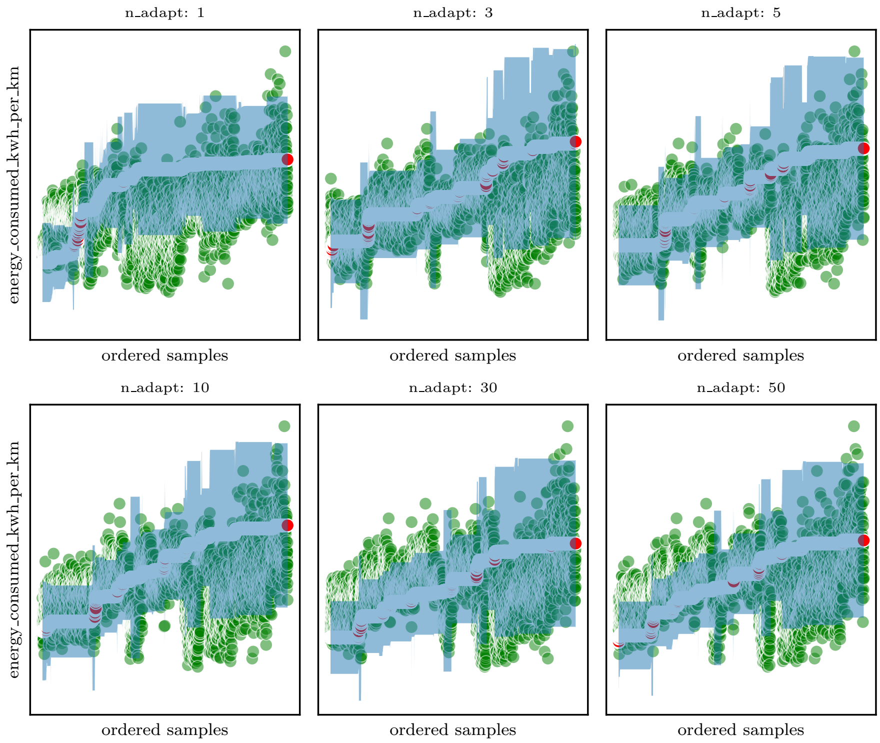


Figure 4.8: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 3 by the BNN model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty.

After 50 adaptation samples the Wasserstein metric demonstrates promising performance with a score of 0.89 as well as the histogram of CDF values in Figure 4.9 being close to uniform. However, a closer examination of the sub-figure corresponding to 50 adaption samples in Figure 4.8 reveals a disparity in performance. Notably, the sub-figure illustrates a trend of underprediction on target domain data points that the model predicts low energy consumption for, and overprediction on data points that the model predicts high energy consumption for. This results in a offsetting of CDF values which might be the reason for the nearly uniform histogram.

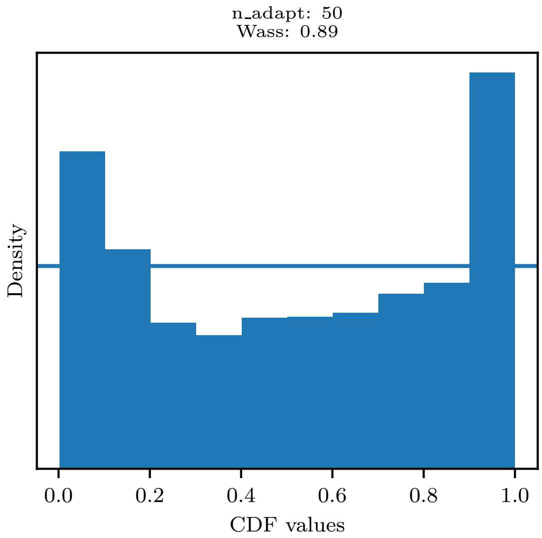


Figure 4.9: Histogram of CDF values by the BNN model's predictions on Case 3 adapted on 50 adapation samples.

4.4 Case 4 - New Vehicle Manufacturer

In the following sections, we will present the results from conducting the experiment on Case 4. This Case is characterized by multivariate covariate shift. Details are available in Section 3.2.4. Table 4.6 presents the performance metrics from the conducted experiment on Case 4.

4.4.1 Baseline Models

The results for the source-trained baseline model when tested on the target domain evaluation dataset without having been introduced to any adaptation samples are among the worst out of any model and Case. The situation that the model finds itself in is that during during training on the source data, no fitting of the coefficient of the target domain vehicle manufacturer is made. Then, when introduced to the target data and the third vehicle manufacturer, the coefficient is wrong, which seems to have a big impact on the predictions. However, only one adaptation sample is needed for the model to perform drastically better on the target test data.

The initially bad performance on the target test data is due to underprediction, see left-most sub-figures in Figure 4.10. As stated, the source-trained model then quickly adapts to the target data but makes too broad predictions, evident by the wide prediction interval in the top sub-figures and the pyramid shaped histograms in the bottom sub-figures of Figure 4.10. The source-trained baseline model show more signs of model adaptation on this Case than for other Cases, specifically regarding individual predictions on target test data points. Where as in previous Cases next to no change is visible, on this Case the model outputs different predictions after almost every adaptation stage.

Metric	Model	Adaptation Samples 0	1	3	5	10	30	50
CRPS	ST	0.18 ± 0.0	0.08 ± 0.02	0.06 ± 0.01	0.06 ± 0.01	0.05 ± 0.01	0.05 ± 0.0	0.05 ± 0.0
	TO	n.a.	n.a.	n.a.	0.07 ± 0.04	0.08 ± 0.02	0.03 ± 0.0	0.03 ± 0.0
	HLR	0.21 ± 0.0	0.10 ± 0.02	0.07 ± 0.01	0.06 ± 0.01	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0
	GPR	0.12 ± 0.0	0.09 ± 0.02	0.07 ± 0.02	0.06 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.04 ± 0.01
	BNN	0.32 ± 0.0	0.14 ± 0.09	0.16 ± 0.06	0.19 ± 0.07	0.23 ± 0.04	0.25 ± 0.03	0.22 ± 0.02
MAPE	ST	0.25 ± 0.0	0.10 ± 0.03	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.07 ± 0.0	0.07 ± 0.0
	TO	n.a.	n.a.	n.a.	0.09 ± 0.02	0.08 ± 0.01	0.07 ± 0.0	0.07 ± 0.0
	HLR	0.16 ± 0.0	0.10 ± 0.03	0.08 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.07 ± 0.0	0.07 ± 0.0
	GPR	0.10 ± 0.0	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.07 ± 0.0	0.07 ± 0.0
	BNN	0.24 ± 0.0	0.16 ± 0.04	0.16 ± 0.01	0.17 ± 0.02	0.18 ± 0.01	0.19 ± 0.01	0.18 ± 0.01
R^2	ST	neg.	neg.	neg.	0.00 ± 0.26	0.06 ± 0.14	0.17 ± 0.01	0.17 ± 0.02
	TO	n.a.	n.a.	n.a.	neg.	0.10 ± 0.17	0.22 ± 0.07	0.25 ± 0.03
	HLR	neg.	neg.	neg.	0.04 ± 0.16	0.13 ± 0.14	0.25 ± 0.06	0.26 ± 0.03
	GPR	neg.	neg.	neg.	0.06 ± 0.14	0.09 ± 0.19	0.22 ± 0.09	0.27 ± 0.05
	BNN	neg.	neg.	neg.	neg.	neg.	neg.	neg.
Wass	ST	neg.	0.64 ± 0.12	0.73 ± 0.07	0.76 ± 0.07	0.78 ± 0.05	0.83 ± 0.01	0.83 ± 0.01
	TO	n.a.	n.a.	n.a.	0.45 ± 0.06	0.64 ± 0.06	0.91 ± 0.04	0.93 ± 0.04
	HLR	0.45 ± 0.0	0.58 ± 0.04	0.68 ± 0.04	0.72 ± 0.04	0.75 ± 0.03	0.80 ± 0.01	0.81 ± 0.01
	GPR	0.57 ± 0.0	0.64 ± 0.05	0.71 ± 0.05	0.75 ± 0.03	0.78 ± 0.05	0.84 ± 0.02	0.85 ± 0.01
	BNN	neg.	0.65 ± 0.19	0.79 ± 0.13	0.77 ± 0.11	0.81 ± 0.11	0.75 ± 0.09	0.82 ± 0.07

Table 4.6: Mean ± standard deviation of Wasserstein score, CRPS, MAPE and R^2-score across 30 iterations on the target evaluation dataset of Case 4, for each model adapted on various number of adaptation samples. For each size of adaptation samples and metric, the best score is highlighted.

The results for the target-only baseline model in Table 4.6 indicate large improvements to probabilistic calibration as the model is adapted on adaptation samples from the target domain, achieving the highest Wasserstein score as well as lowest CRPS score after 50 adaptation samples. After only 5 adaptation samples, and also 10, 30 and 50 adaptation samples, the target-only model accomplishes better MAPE score than in other Cases after 50 adaptation samples. On the other hand the R^2-score that the model achieves is low at 0.25 after 50 adaptation samples.

4.4.2 Hierarchical Bayesian Linear Regression Model

Large improvements are made to model performance in regards to all measurements, leading to the best average MAPE after 50 adaptation samples among all Cases for the hierarchical model. On a high level, not relative to the other models in this Case, the final probabilistic calibration is good, with a CRPS of 0.05 and Wasserstein score of 0.81.

In general, the hierarchical model outputs too broad predictions, especially when not having been adapted on any adaptation samples. As more adaptation samples are introduced to the model the prediction interval shrinks, but the prediction uncertainties are still too broad after 50 adaptation samples, as indicated by the histogram of CDF values in Figure 4.11. However, looking at the prediction interval

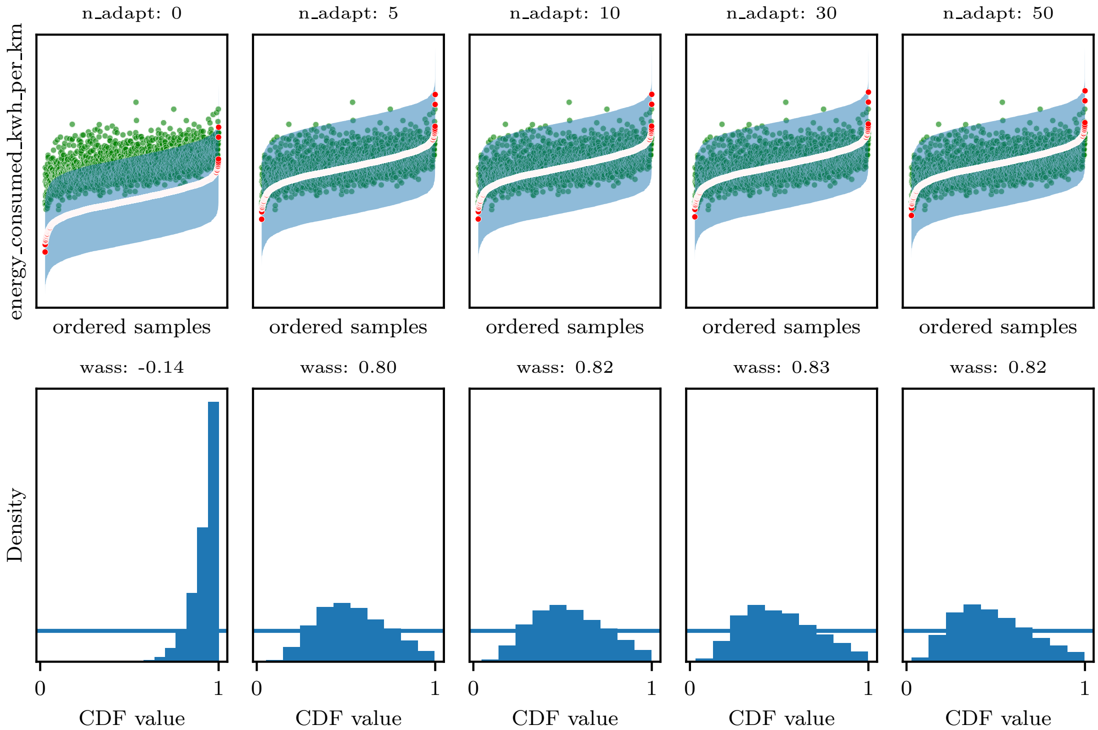


Figure 4.10: **Top row**: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 4 by the source-trained baseline model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty. **Bottom row**: Histograms of CDF values for the same predictions.

plot for `n_adapt = 50` in the top row of the same figure, the prediction interval covers almost all true label values.

4.4.3 Gaussian Process Regression Model

The GPR model achieves the best results before adaptation. The GP model further improves its performance on the target domain evaluation data after adaptation, leading to similar results as the hierarchical and source-trained baseline model based on the non-probabilistic measurements, see Table 4.6. However, the GP model accomplishes better probabilistic calibration, indicated by the Wasserstein score and CRPS.

The reason for the poor probabilistic calibration after 0 adaptation samples is that the GP model produces too broad predictions, i.e. too large standard deviations. This can be seen in both left-most sub-figures in Figure 4.12. The figure also highlights the fact that the model produces too broad predictions throughout the whole adaptation process, according to the histograms of CDF values. Conversely, looking at the upper, right-most two sub-figures they seem to indicate that the prediction interval of the model is not too broad, as it covers all target domain evaluation data points almost perfectly. For individual predictions, the model outputs smaller predicted

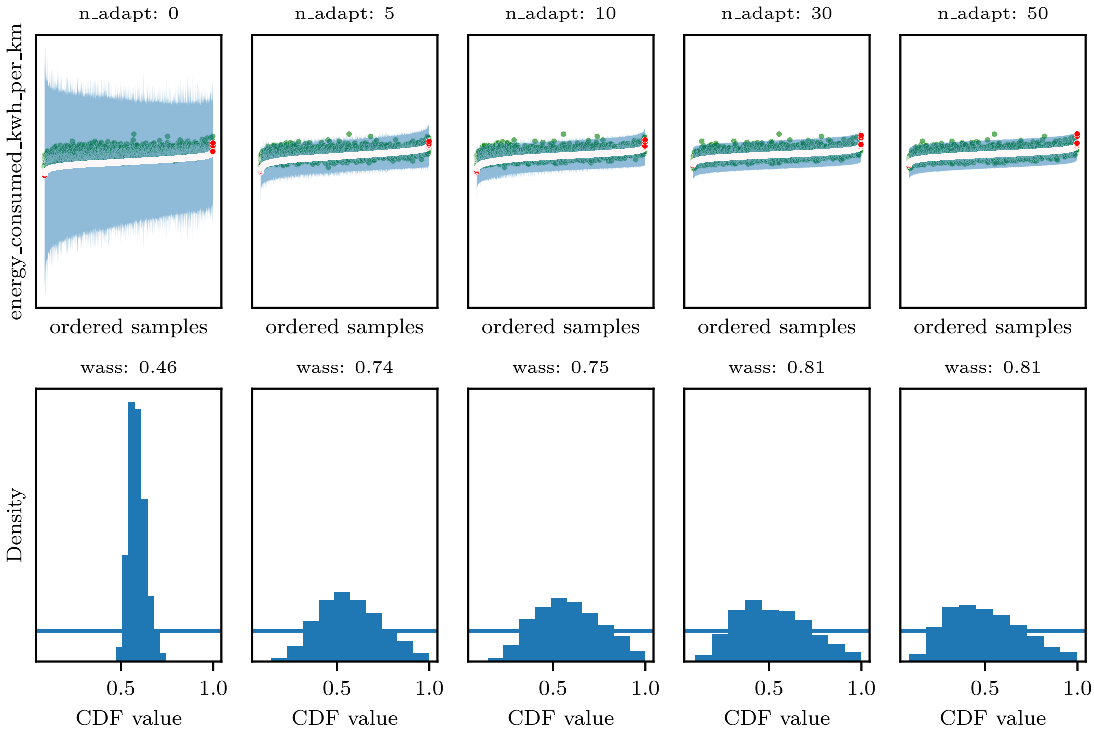


Figure 4.11: **Top row**: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 4 by the HLR model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty. **Bottom row**: Histograms of CDF values for the generated predictions.

standard deviations as it gets adapted on more adaptation samples from the target dataset. Also shown in Figure 4.12 is that the GP model is more uncertain when it predicts low or high energy consumption, regardless of how many adaptation samples the model has been adapted on.

4.4.4 Bayesian Neural Network Model

The averaged results for the Bayesian neural network (BNN), see Table 4.6, indicate that the model performs worse than all the other models on this Case. In particular the R^2-score is low and even negative during the whole adaptation process which means that the model does not properly manage to explain the dependency between the driving conditions and energy consumption. Additionally, the BNN model achieves an abnormally high average CRPS, compared to the other models evaluated on this Case, if adapted on 50 adaptation samples.

Table 4.6 reveals a notable dispersion in the Wasserstein and R^2 metrics. This variability suggests that the BNN model's sensitivity to the specific subset of data used for adaptation significantly influences the predictive performance of the model. For instance, the dispersion of the Wasserstein score first increases then decreases with the number of adaptation samples, indicating no stabilization in model performance by

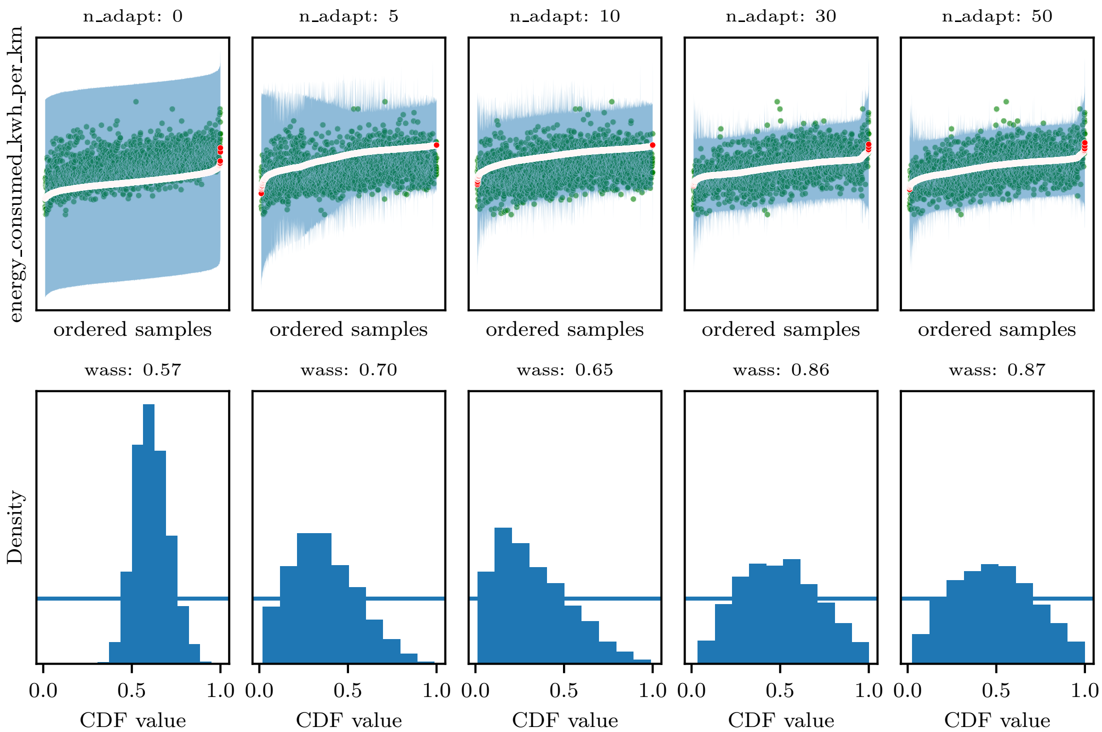


Figure 4.12: **Top row**: Prediction interval plots. Scatter plots of ordered predictions (red, in the middle of the interval) and true labels (green) for Case 4 by the GPR model adapted on various sizes of adaptation samples. The interval (blue) is the 95% prediction interval derived from the estimated prediction uncertainty. **Bottom row**: Histogram of CDF values for the generated predictions.

adaptation. Similarly, the R^2 metric exhibits a high degree of variability, particularly with few adaptation samples. This pattern underscores the importance of the data selection in the adaptation process of the BNN, as the choice of adaptation samples can drastically affect the outcome, potentially leading to either underfitting or overfitting depending on the representativeness of the selected samples.

Figure 4.13 reveal no clear trend towards greater certainty in the BNN model´s predictions for individual data points with increasing adaptation samples, as for other models. The predictive standard deviations do not consistently decrease; instead, they fluctuate. Surprisingly, a prevailing trend towards increased uncertainty is observed as the model encounters a greater number of adaptation samples from the target dataset, regardless of whether the true energy consumption resides within the tail or mode of the target domain label distribution. This phenomenon could indicate that the model, while adapting to new data, might be encountering complexities or variations in the data that it has not previously learned to generalize, thereby increasing the overall uncertainty in its predictions.

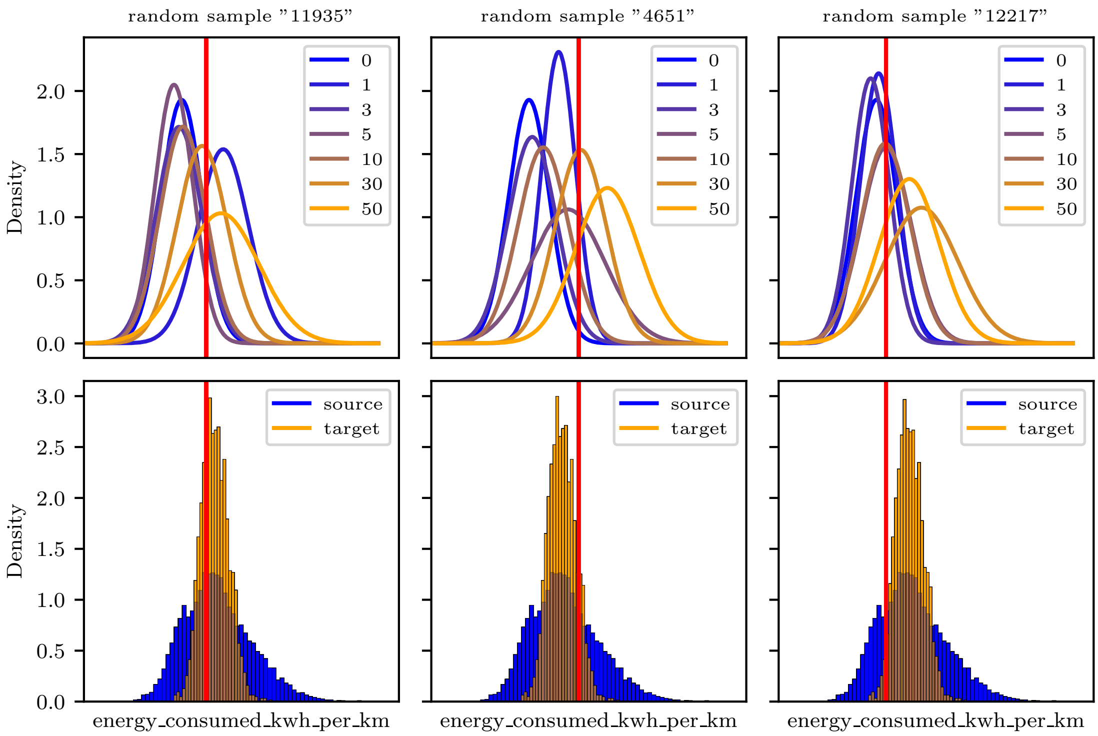


Figure 4.13: **Upper graphs:** Case 4, the BNN model's predictions (Gaussian distributions) for three different `drives` in the target domain evaluation set, across runs with increasing number of adaptation samples (blue to orange). **Lower graphs:** the label distribution of the source (blue) and target domain (orange) respectively. **Vertical line** (red)**:** Illustrates the true label.

5

Discussion

In this chapter, we delve into an analysis of the results obtained from the various models and Cases studied. The aim of this thesis was to assess the characteristics and efficacy of different machine learning models in addressing *covariate shift* arising from new operational conditions for electric trucks. This involved evaluating the robustness of these models in providing accurate probabilistic predictions for new driving conditions, given limited data from those environments. The following paragraphs will discuss the strengths and limitations of each model, thereby providing insights that can inform future research and practitioners in the field of electric truck fleet management.

5.1 Baseline Models

The *source-trained baseline model*'s results on the Cases provide interesting insights. The model handles the covariate shift in Case 1 (Temperature Univariate Shift) the best, even before adaptation, and industry knowledge suggest the model has learned an accurate representation of temperature's dependency on energy consumption for lower temperatures. An accurately learned representation is a characteristic shared with the *target-only baseline model* when adapted on 30 or more samples, which also performs well as measured by how accurately centered the predictions are compared to the true labels, low MAPE.

Despite this, the target-only baseline model does not reach the same level of probabilistic calibration in its predictions, indicating that knowledge of the source domain improves uncertainty quantification in the target domain. Interestingly, the opposite holds true for Case 3 (New Type of Route), which is characterized by a bivariate covariate shift. In this Case, the target-only baseline model performs well and instead the source-trained baseline model suffers from prior training on the source domain combined with low adaptation sensitivity (rate of adaptation). This is inferred by the fact that adaptation has a slightly larger effect on ST, relative to Case 1, suggesting that the learned source representation differs from the target's.

For the Cases exhibiting multivariate covariate shift in Case 2 (Temperature Multivariate Shift) and Case 4 (New Vehicle Manufacturer), the predictive distributions generated by the source-trained baseline model are considerably worse before adaptation, especially for Case 4. After adaptation, the predictive distributions improve and result in competitive MAPE scores, but the level of probabilistic calibration remains inferior. As illustrated in Figure 4.10, the uncertainties inferred by the source-trained model are overestimations. Conversely, the target-only baseline model is consistently, for large adaptation sample sizes, the model generating the most probabilistically accurate predictions for the multivariate covariate shift Cases, although quite variable.

The overestimation of prediction uncertainty by the source-trained baseline model could be explained by the Frequentist approach and the uncertainty estimation technique employed, as detailed in Theory 2.1.1.2, where a portion of the estimated uncertainty is derived from the variation in the source (training) data. Consequently, given the Frequentist approach, when the observations of the target domain are few compared to the source domain, and the target domain exhibit less variability, the predicted uncertainty is overestimated, caused by the source domain's more pronounced variability. Conversely, the target-only baseline model which exclusively observe the target domain, naturally captures the relevant variability. Notably, the target-only baseline model requires at least 30 or more samples from the target domain to reach this performance.

5.2 Hierarchical Bayesian Linear Regression Model

It is evident that the adaptation behavior of the Hierarchical Bayesian Linear Regression (HLR) model differs based on the nature of the covariate shift. In instances where the shift occurs in continuous variables (Case 1-3), the prediction uncertainties remains relatively unchanged across adaption rounds, but the mean of the predicted Gaussian distributions changes. This implies that the model's adaptation primarily results in shifts in the mean of its predictive distributions to align with the target domain data, rather than updated beliefs about the width of the predictive distributions when covariate shift in continuous features are present.

Conversely, in Case 4, when the covariate shift affects the hierarchical level of the HLR model, the adaptation of the model is more pronounced, which is to be expected. The prediction means and the prediction uncertainties undergo significant changes, indicating a larger adjustment in the model's parameters to effectively capture the shifted distribution. Initially, the model exhibits substantial uncertainty in its predictions upon introduction to the target domain, as seen in the left-most sub-figures of Figure 4.11. However, as it adapts and learns from the adaptation samples, the prediction interval is narrowed already after 5 adaptation samples, leading to improved probabilistic calibration. The greater adaptation sensitivity displayed by the HLR model in Case 4 can be explained by the model's Bayesian intrinsics.

The HLR model's strength in utilizing knowledge from various data groupings, within its hierarchical structure, to predict outcomes for new groups of data is especially evident in Case 4 (New Vehicle Manufacturer). The target domain drives can be treated as a new grouping of data modelled by parameters $\boldsymbol{\theta}$, coming from the posterior distribution of the across-groupings, shared hyperparameter $\boldsymbol{\mathcal{E}}$. Hence, the prior beliefs of $\boldsymbol{\theta}$ will be based on dependencies learnt from vehicle manufacturers included in the source domain, but as initially no evidence has been incorporated to those beliefs, the prediction uncertainties are large. Subsequently, when the prior of $\boldsymbol{\theta}$ is updated with adaptation samples, the uncertainty about the parameters is reduced and consequentially the prediction uncertainties as well, which explains the large exhibited adaptation sensitivity in Case 4. This behavior suggests that the HLR model is better equipped to handle covariate shift in categorical features that can be modelled as hierarchical levels, compared to covariate shift in continuous variables that in essence can not.

Regarding model performance, it is noteworthy that even without any adaptation samples the HLR model, compared to other models, exhibits strong probabilistic calibration across all Cases, especially for those where the covariate shift is over continuous variables. This is reflected in consistently good CRPS and Wasserstein score across the Cases, before adaptation, underscoring a capacity to learn a representation, $f(\boldsymbol{x}; \theta_{\mathrm{S}}) = \hat{p}(y|\boldsymbol{x}) \approx p_{\mathcal{T}}(y|\boldsymbol{x})$, by training on the source domain that transfers well to the target domain. This suggest that the relationships between some of the individual features and energy consumption are linear, which is confirmed by industry knowledge. Hence, the Bayesian HLR model displays robustness by mitigating, to a larger extent, the various source-training biases exposed by covariate shift, compared to the other models.

5.3 Gaussian Process Regression Model

The Gaussian Process Regression (GPR) had a tendency to overestimate the prediction uncertainties. This is illustrated in the prediction interval plots, where most points fall within the 95% prediction interval. Although, generally, the models is not probabilistically calibrated, the model's conservative uncertainty estimation is at least preferable compared to the opposite in fleet management operations.

The model's performance was significantly bad when evaluated on the target domain before adaptation, indicating an inability to learn the true conditional distribution from the source domain. This limitation probably stems from the GPR model's *generative* characteristic. Additionally, the GPR model shows slightly worse performance on Cases with multivariate covariate shift. This reduced performance may be due to the selection of the Matern kernel as a covariance function when dealing with data where multiple features are shifted. We propose for further research to investigate whether alternative kernels can better handle the effects of multivariate covariate shift.

However, after adaptation on 50 target domain drives, the model achieved the highest R^2-scores of all models across all Cases, highlighting its responsiveness to adaptation. Intuitively, the results after 50 adaptation samples suggests that the model has learned an accurate representation of the target domain. On the contrary, the generative GPR model might simply model the target domain data well, but deviate significantly from the true conditional distribution. Therefore, due to its generative nature, the GPR model is particularly sensitive to the subset of data from the target domain that it has been adapted to.

5.4 Bayesian Neural Network Model

The Bayesian Neural Network (BNN) model's performance across the various types of covariate shift examined in the Cases, indicates that while the model succeed to estimate prediction uncertainty quite well, it struggles with prediction mean accuracy and model fit.

The BNN model's performance across all Cases demonstrates a critically high level of adaptation sensitivity, specifically to the subset of data used for adaptation. For instance, adaptation based on a single sample, as opposed to none, often results in worse model predictions. In the left-hand graphs of Figure 4.2 and 4.3 it is evident that the flat temperature slope learned for the source domain is quite wrong, supported by industry knowledge. Interestingly, the slope for the target domain is more accurate. However, as previously discussed, the model is highly sensitive to adaptation, as shown in the right-hand graphs of both figures. Consequently, the adaptation technique of the model, despite being theoretically sound, may be insufficient in practice without a larger and more representative set of data. Further analysis is needed to determine whether additional source data would stabilize the adaptation process or if the limited amount of target domain samples, the Few-Shot setup, is inherently problematic for a neural network.

6

Conclusion

Fleet managers seeking to expand their operations while maintaining operational efficiency demand accurate and informative energy consumption modeling. New driving conditions caused by expansion present unique challenges by inducing *covariate shift* to the data used for modeling. The thesis aimed to address these challenges. Our investigation focused on four distinct Cases characterized by conditions realistic for fleet managers in the logistics sector, each representing a unique scenario of covariate shift caused by colder temperatures, new types of routes or new vehicle manufacturers. By evaluating the probabilistic predictions of different machine learning models, adapted on data collected from *test-drives* in new driving conditions, significant insights were gained into the models' robustness and adaptability. Overall, the thesis highlights the importance of model selection and adaptation strategies in achieving informative, reliable and accurate energy consumption predictions for new driving conditions.

The *source-trained linear baseline model* demonstrated consistent ability to infer dependencies from the source domain, which translated well to target domains exhibiting uni- or bivariate covariate shift. However, for multivariate covariate shift, where the dependencies learned from the source domain deviates to wider extent compared to those inferred from the target domain, the model struggles to adapt due to low adaptation sensitivity. This resulted in poorer performance compared to the *target-only linear baseline model*. However, the target-only model requires a large set of adaptation samples, 30 or more, to generate accurate predictive distributions.

The Hierarchical Bayesian Linear Regression model consistently demonstrated strong probabilistic calibration and adaptability, particularly for Case 4 (New Vehicle Manufacturer) where the covariate shift stems from a categorical feature, modeled hierarchically. This model's ability to utilize hierarchical structures within the data allowed it to effectively adjust to new vehicle manufacturers. Whether utilization of hierarchical structures could improve non-linear models is suggested for future research.

The Gaussian Process Regression model initially overestimated prediction uncertainties. Adaptation reduced overestimation, yet it highlighted a potential sensitivity to the selection of kernel depending on the type of covariate shift. Future research should explore alternative kernels to better handle multivariate covariate shifts and enhance probabilistic calibration.

The Bayesian Neural Network model, despite its theoretical advantages in estimating prediction uncertainty, struggled with accurately estimating the prediction mean. The BNN model demonstrated high sensitivity to the available target domain data, often resulting in fluctuating prediction uncertainties. Future work should investigate whether the FSDA setup with few data samples is inherently challenging for BNNs, or if alternative adaptation techniques could address the observed limitations.

Bibliography

[1] P. W. Koh, S. Sagawa, H. Marklund, *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 5637–5664. [Online]. Available: `https://proceedings.mlr.press/v139/koh21a.html`.

[2] Y. Chen, G. Wu, R. Sun, A. Dubey, A. Laszka, and P. Pugliese, "A review and outlook of energy consumption estimation models for electric vehicles," *arXiv preprint arXiv:2003.12873*, 2020.

[3] C. Fiori and V. Marzano, "Modelling energy consumption of electric freight vehicles in urban pickup/delivery operations: Analysis and estimation on a real-world dataset," *Transportation Research Part D: Transport and Environment*, vol. 65, pp. 658–673, 2018.

[4] A. D. Alvarez, F. S. Garcia, J. E. Naranjo, J. J. Anaya, and F. Jimenez, "Modeling the driving behavior of electric vehicles using smartphones and neural networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 3, pp. 44–53, 2014.

[5] D. Jiménez, S. Hernández, J. Fraile-Ardanuy, J. Serrano, R. Fernández, and F. Alvarez, "Modelling the effect of driving events on electrical vehicle energy consumption using inertial sensors in smartphones," *Energies*, vol. 11, no. 2, p. 412, 2018.

[6] E. Yao, M. Wang, Y. Song, and Y. Zhang, "Estimating energy consumption on the basis of microscopic driving parameters for electric vehicles," *Transportation Research Record*, vol. 2454, no. 1, pp. 84–91, 2014.

[7] R. Zhang and E. Yao, "Electric vehicles' energy consumption estimation with real driving condition data," *Transportation Research Part D: Transport and Environment*, vol. 41, pp. 177–187, 2015.

[8] J. Alves, P. Baptista, G. Gonçalves, and G. Duarte, "Indirect methodologies to estimate energy use in vehicles: Application to battery electric vehicles," *Energy conversion and management*, vol. 124, pp. 116–129, 2016.

[9] W. Li, P. Stanula, P. Egede, S. Kara, and C. Herrmann, "Determining the main factors influencing the energy consumption of electric vehicles in the usage phase," *Procedia Cirp*, vol. 48, pp. 352–357, 2016.

[10] K. Hu, J. Wu, T. Schwanen, *et al.*, "Differences in energy consumption in electric vehicles: An exploratory real-world study in beijing," *Journal of Advanced Transportation*, vol. 2017, 2017.

[11] S. Sun, J. Zhang, J. Bi, Y. Wang, *et al.*, "A machine learning method for predicting driving range of battery electric vehicles," *Journal of Advanced Transportation*, vol. 2019, 2019.

[12] G. M. Fetene, S. Kaplan, S. L. Mabit, A. F. Jensen, and C. G. Prato, "Harnessing big data for estimating the energy consumption and driving range of electric vehicles," *Transportation Research Part D: Transport and Environment*, vol. 54, pp. 1–11, 2017.

[13] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. [Online]. Available: `http://probml.github.io/book2`.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[15] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 69, no. 2, pp. 243–268, 2007.

[16] F. Wick, U. Kerzel, M. Hahn, M. Wolf, T. Singhal, and M. Feindt, "Demand forecasting of individual probability density functions with machine learning," *CoRR*, vol. abs/2009.07052, 2020. arXiv: `2009.07052`. [Online]. Available: `https://arxiv.org/abs/2009.07052`.

[17] F. X. Diebold, T. A. Gunther, and A. S. Tay, "Evaluating density forecasts with applications to financial risk management," *International Economic Review*, vol. 39, no. 4, pp. 863–883, 1998. DOI: `10.2307/2527342`.

[18] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[19] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009, p. 248, ISBN: 9780262170055.

[20] M. D. Hoffman, A. Gelman, *et al.*, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo.," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[21] W. J. Maddox, S. Stanton, and A. G. Wilson, "Conditioning sparse variational gaussian processes for online decision-making," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6365–6379, 2021.

[22] B. Lucas, C. Pelletier, D. Schmidt, G. I. Webb, and F. Petitjean, "A bayesian-inspired, deep learning-based, semi-supervised domain adaptation technique for land cover mapping," May 2020, Accessed: 2024-05-10. arXiv: `2005.11930 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2005.11930`.

[23] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012, ISBN: 9780262018029.