UNIVERSITY OF CALIFORNIA,
IRVINE

Multi-model Online Conformal Prediction

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering and Computer Science

by

Erfan Hajihashemi

THESIS Committee:
Assistant Professor Yanning Shen, Chair
Chancellor's Professor Syed Ali Jafar
Associate Professor Yasser Shoukry

2025

DEDICATION

I dedicate this work to my beloved grandfather, Hossein Hajihashemi, who passed away while I was away from home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Yanning Shen, for her continuous support of research, as well as her patience, motivation, enthusiasm, and immense knowledge.

Beside my advisor, I would also like to extend my heartfelt thanks to my committee members, Prof. Syed Ali Jafar and Prof. Yasser Shoukry, not only for dedicating their valuable time to review my work but also for their insightful feedback, which pushed my research to a higher level.

Last but not least, I would like to thank my family: my parents, Amir and Leila, for their unwavering support; my aunt, Sheida, for always being so supportive and helpful here in the U.S.; and finally, my only brother, Aref, for always being there for me.

VITA

Erfan Hajihashemi

EDUCATION

Master of Science in Electrical Engineering and Computer Science **2025**
University of California, Irvine *Irvine, State*

Bachelor of Science in Electrical and Computer Engineering **2023**
University of California, Irvine *Tehran, Iran*

RESEARCH EXPERIENCE

Graduate Research Assistant **2023–2025**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2024–2025**
University of California, Irvine *Irvine, California*

REFEREED CONFERENCE PUBLICATIONS

Erfan Hajihashemi, Yanning Shen, "Multi-model Ensemble Conformal Prediction in Dynamic Environments," Neural Information Processing Systems (**NeurIPS**), 2024.

ABSTRACT OF THE THESIS

Multi-model Online Conformal Prediction

By

Erfan Hajihashemi

Master of Science in Electrical Engineering and Computer Science

University of California, Irvine, 2025

Assistant Professor Yanning Shen, Chair

Many machine learning algorithms face challenges in predicting labels with high certainty, falling short of achieving the desired levels of accuracy and other critical evaluation metrics. This necessitates predicting a set of candidate labels with a valid coverage probability of the true label, rather than limiting to a single label. Conformal prediction is an uncertainty quantification method that constructs a prediction set for a previously unseen datum, ensuring the true label is included in the set with a predetermined coverage probability. Adaptive conformal prediction has been developed to address data distribution shifts in dynamic environments. However, the efficiency of the prediction set (i.e., prediction set size) varies depending on the learning model used. Employing a fixed model may not consistently offer the best performance in dynamic environments with unknown data distribution shifts. To address this issue, we introduce a novel adaptive conformal prediction framework, where the model used for creating prediction sets is selected 'on the fly' from multiple candidate models. The proposed algorithm is proven to achieve strongly adaptive regret over all intervals while achieving a coverage probability aligned with the target value. Experiments on real and synthetic datasets corroborate that the proposed approach consistently yields more efficient prediction sets while including the true label in the prediction set with the desired probability, outperforming alternative methods.

Chapter 1

Introduction

Most machine learning algorithm designs aim to enhance label prediction accuracy. Nevertheless, a significant challenge persists, as many models demonstrate limitations in predicting labels with high certainty, falling short of achieving the desired levels of accuracy and other critical evaluation metrics. In applications such as medical diagnosis, it is sometimes more efficient to predict a subset of labels rather than a single label [24, 33]. This necessitates predicting a set of candidate labels with a valid coverage probability, rather than limiting to a single label. Such prediction sets should ideally balance coverage and compactness to facilitate reliable and interpretable decision-making.

One of the most widely used frameworks for set prediction is conformal prediction [39]. Conventional conformal prediction algorithms can achieve the desired coverage assuming the exchangeability of data [2]. A sequence of random variables $z_1, \ldots, z_N$ is said to be exchangeable if the joint distribution is invariant under any permutation of the indices. That is, any reordering of the sequence is equally likely. This condition is weaker than the assumption of independence and identical distribution (i.i.d.). While all i.i.d. sequences are exchangeable, the converse does not hold [31]. Exchangeability guarantees that the

variables have the same marginal distribution, but they are not necessarily independent. In many real-world online problems [20], the distribution of data shifts over time, making the exchangeability assumption no longer applicable. Consequently, adaptive conformal prediction algorithms have been developed [13], where prediction sets are constructed in a time-varying manner by incorporating the most recent data to recalibrate the coverage guarantees.

Despite these advancements, the efficiency (e.g., prediction set size or regret) of online conformal prediction methods heavily depends on the model employed, and a single model may not consistently perform well across various distribution shifts. In particular, using a fixed learning model throughout the entire horizon may lead to degraded performance. Creating an efficient prediction set size is as important as obtaining the desired coverage. Trivial cases may arise where the conformal prediction algorithm may alternate between producing an empty prediction set and a full label set, such that the full set is output with probability equal to the target coverage level, and the empty set is output otherwise. While this strategy technically satisfies the desired coverage guarantee, it yields impractically large prediction sets and lacks meaningful informativeness [5]..

To address this limitation, our proposed algorithm incorporates multiple learning models simultaneously. It dynamically selects the suitable model based on the performance of each model with the most recently received data. The selection is informed by a feedback mechanism that evaluates conformity scores across models, enabling the algorithm to adaptively favor those better aligned with the current data distribution.

Related work: Conformal prediction [39, 31, 38] is an effective method for uncertainty quantification that has been widely used to predict a set of candidate labels for incoming data. It treats the learning model as a black box and produces a prediction set for each test instance. Conformal prediction is applicable to both classification [9, 32, 30] and regression [29, 28, 6] tasks.

In dynamic environments, where the data distribution shifts over time, applying standard conformal prediction algorithms may fail to achieve the desired coverage guarantees. To address this challenge, conformal prediction under distribution shift has been studied recently. [36] explored conformal prediction in dynamic settings using a reweighting approach; however, their method requires prior knowledge of the data's dependency structure. [3] resolved this issue by requiring the weights to remain fixed.

The incorporation of time-varying coverage probability was introduced in [13], although choosing an appropriate step size remains a significant challenge. One promising approach to handling dynamic environments involves drawing from the literature on learning with expert advice [7, 37, 26]. [40] suggested that tuning the step size through expert (base learner) aggregation could be effective. However, their method assigns equal importance to all historical data, which limits the algorithm's adaptability to sharp distributional changes. [14] addressed this issue by employing multiple experts, each associated with a different step size from a pool of candidates, leading to varying coverage levels at each time step. While this method demonstrates adaptive regret across time intervals of fixed width, [5] showed that it does not guarantee suitable regret bounds for arbitrary interval lengths. To overcome this, [5] proposed a method that achieves strongly adaptive regret [8] across arbitrary time intervals by assigning a specific interval to each expert. Although their algorithm achieves sublinear regret, it depends on hyperparameter choices that control the lifetime of each expert.

The approach proposed in this study similarly assigns specific time intervals to experts. However, unlike previous methods, each expert in the proposed framework contains multiple learning models and selects the appropriate model according to the prevailing data distribution. This allows the method to construct more efficient prediction sets while maintaining valid coverage. It is worth noting that the materials in this thesis are included in [15]

Contributions. Overall, our contributions can be summarized as follows:

I) We introduce a novel adaptive conformal prediction algorithm, **S**trongly **A**daptive **M**ultimodel Ensemble **O**nline **C**onformal **P**rediction (SAMOCP), designed for dynamic environments with unknown distribution shifts. This algorithm incorporates multiple models and dynamically selects a model based on its performance in previous time steps.

II) We demonstrate that SAMOCP exhibits strongly adaptive regret for any arbitrary time interval while ensuring valid coverage.

III) Through experimental tests on classification tasks subject to distribution shifts, we demonstrate that SAMOCP outperforms existing methods by constructing more efficient prediction sets while also achieving a coverage probability closely aligned with the target value.

Chapter 2

Preliminaries

2.1 Standard Conformal Prediction

This section explains standard conformal prediction in the context of adaptive online settings, where data arrives sequentially.

Conformal prediction provides a principled approach for uncertainty quantification by wrapping around any black-box model and outputting a set of plausible labels instead of a single prediction. The key idea is to assign a score—called a non-conformity score—to each possible label, measuring how well it fits the input relative to the historical data. A candidate label is included in the prediction set if its score is sufficiently low, meaning it conforms with past observations.

The method relies on the assumption that data points are exchangeable, so the scores from past examples can be used to define a valid threshold for new, unseen inputs. If the current label's score is among the lowest $(1-\alpha)$ fraction of all scores, it is considered plausible. This way, the prediction set guarantees a user-defined coverage level, meaning the true label is

included in the set with high probability (at least $1-\alpha$)—regardless of the underlying data distribution or model type.

Given a miscoverage probability α, a learning model m, a historical dataset $\{(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t-1}$, and a new input $X_t \in \mathcal{X}$, the objective is to construct a prediction set $C_\alpha^m(X_t) \subseteq \mathcal{Y} := \{1, 2, \ldots, K\}$, where K denotes the total number of classes, such that $C_\alpha^m(X_t)$ contains the true label Y_t^{true} with probability at least $1-\alpha$.

In the online setting, the historical dataset is updated to $\{(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t}$ at the end of each time step t, when the true label Y_t^{true} for input X_t is observed. In this scenario, conformal prediction treats the historical dataset as a calibration dataset, which is used to determine whether a candidate label $Y \in \mathcal{Y}$ should be included in the prediction set.

Consequently, conformal prediction operates in an online fashion using an evolving calibration dataset—consisting of all previously observed labeled samples—to construct a valid prediction set. A set of non-conformity scores $\{S^m(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t-1}$ is defined, where each score $S^m(X_\tau, Y_\tau^{\text{true}})$ quantifies the disagreement between the true label Y_τ^{true} and the predicted label $\hat{f}^m(X_\tau)$.

Given a new input X_t, the standard conformal prediction algorithm constructs the prediction set as

$$C_\alpha^m(X_t) = \{Y \in \mathcal{Y} \mid S^m(X_t, Y) \leq \hat{q}_\alpha^m\},$$

where the threshold $\hat{q}_\alpha^m$ is determined by the quantile of past non-conformity scores:

$$\hat{q}_\alpha^m = \text{Quantile}\left(\frac{\lceil t(1-\alpha)\rceil}{t-1}, \{S^m(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t-1}\right). \tag{2.1}$$

The Quantile function sorts all non-conformity scores of the historical data and then identifies the $\lceil t(1-\alpha)\rceil$th smallest score as $\hat{q}_\alpha^m$.

2.2 Adaptive Conformal Prediction

In standard conformal prediction, the miscoverage probability $1-\alpha$ is fixed over time. While this assumption enables formal guarantees under the exchangeability condition, it can lead to overly conservative or under-confident prediction sets when applied to data streams that exhibit distribution shift. In particular, when the underlying data distribution changes, a fixed threshold may no longer reflect the current data characteristics.

To address this challenge, adaptive conformal prediction methods have been proposed, allowing the miscoverage probability to vary over time. Specifically, the static α in standard conformal prediction is replaced with a time-varying sequence $\{\alpha_t\}_{t=1}^{T}$, where each α_t is updated based on the observed performance of the prediction set at time t. In such settings, the quantile threshold $\hat{q}_{\alpha}^{m}$ from (2.1) becomes

$$\hat{q}_{\alpha_t}^{m} = \text{Quantile}\left(\frac{\lceil t(1-\alpha_t)\rceil}{t-1}, \{S^m(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t-1}\right), \tag{2.2}$$

and α_t is adapted after observing whether the true label Y_t^{true} falls within the prediction set generated at time t.

This form of adaptation allows the algorithm to increase α_t when the prediction set is too conservative (e.g., always covering the true label) and decrease it when coverage is violated too frequently. The resulting feedback loop dynamically calibrates the prediction set to better reflect the shifting data distribution.

Recent studies on online conformal prediction under distribution shift have incorporated adaptive miscoverage probabilities to enhance performance in dynamic environments. Nevertheless, approaches that rely on a single fixed learning model often struggle to maintain consistent efficiency as the distribution evolves. This limitation motivates the use of ensemble or multi-model strategies.

To this end, we propose a novel adaptive multi-model online conformal prediction algorithm designed to identify the most suitable learning model at each time step t. At every time slot, the goal is to construct a prediction set for the new input X_t, using the historical dataset $\{(X_\tau, Y_\tau^{\text{true}})\}_{\tau=1}^{t-1}$, such that the true label Y_t^{true} is included in the prediction set with probability at least $1-\alpha$. The proposed algorithm ensures valid coverage and simultaneously achieves strongly adaptive regret over arbitrary time intervals, enhancing its effectiveness in highly dynamic settings.

Chapter 3

Multi-model Ensemble Conformal Prediction

3.1 Introduction

This chapter presents a framework for multi-model conformal prediction tailored to static environments, where the underlying data distribution remains fixed. The idea is to maintain multiple learning models, each with its own conformal predictor, and adaptively select among them based on historical performance. This approach addresses the inefficiency of relying on a single fixed model, which may produce overly conservative prediction sets in practice.

The remainder of this chapter is organized as follows. Section 3.2 introduces the Multi-model Online Conformal Prediction (MOCP) algorithm, which forms the core of the proposed method. It consists of three components: (i) model-specific calibration of miscoverage probabilities and (ii) probabilistic model selection via weight-based sampling. Together, these components enable the construction of prediction sets that are both valid and efficient in the static setting. In section 3.3, a regret analysis is provided, demonstrating that MOCP

achieves sublinear regret with respect to the best fixed model and coverage level in hindsight.

3.2 Multi-model Conformal Prediction in Static Environment

The MOCP algorithm combines model-specific calibration and adaptive model selection to improve prediction efficiency. First, we adopt an online learning approach to update each model's miscoverage probability using the pinball loss and SF-OGD algorithm. Second, a model selection mechanism based on exponentially weighted sampling is used to choose a miscoverage probability and its corresponding learning model to construct the prediction set at each time step.

3.2.1 Model-Specific Miscoverage Probability Update

Note that the non-conformity score $S^m(X_\tau, Y_\tau^{true})$ depends on the learning model. Such dependency leads to a model-specific ordering of non-conformity scores, yielding different prediction sets for each model. After observing Y_t^{true}, the adaptive miscoverage probability α_t must be updated for time $t+1$ to cope with distribution shifts effectively. Given that different models achieve different prediction sets, assigning and updating the same α_t for different models would be inadequate. Instead, at each time t, we assign a specific miscoverage probability to each model $m \in [M]$, denoted as α_t^m, and update it based on the corresponding prediction set.

Consequently, for M learning models, there are M candidates for miscoverage probability α_t at each time t. Each candidate is updated according to a distinct rule. These M update rules operate in parallel, with each one updating the corresponding miscoverage probability

upon observing the true label. Next, the update procedure for α_t^m will be examined, followed by a detailed explanation of how each instance of MOCP selects the appropriate miscoverage probability from M distinct options at each time step.

To update miscoverage probability α_t^m, we adopt the pinball loss [21], which can be written as

$$L(\bar{\alpha}_t^m, \alpha_t^m) = \alpha(\bar{\alpha}_t^m - \alpha_t^m) - \min\{0, \bar{\alpha}_t^m - \alpha_t^m\}, \tag{3.1}$$

where

$$\bar{\alpha}_t^m = \sup\{\tilde{\alpha} : Y_t^{true} \in C_{\tilde{\alpha}}^m(X_t)\} \tag{3.2}$$

is the best possible value of miscoverage probability for model m at time t, which constructs the smallest prediction set that covers Y_t^{true}.The miscoverage probability α_{t+1}^m can be updated via SF-OGD [27] as

$$\alpha_{t+1}^m = \alpha_t^m - \eta \frac{\nabla_{\alpha_t^m} L(\bar{\alpha}_t^m, \alpha_t^m)}{\sqrt{\sum_{\tau=1}^t \|\nabla_{\alpha_\tau^m} L(\bar{\alpha}_\tau^m, \alpha_\tau^m)\|_2^2}}, \tag{3.3}$$

where η is the learning rate and

$$\nabla_{\alpha_t^m} L(\bar{\alpha}_t^m, \alpha_t^m) = \mathbb{I}[\bar{\alpha}_t^m < \alpha_t^m] - \alpha = err_t^m - \alpha, \tag{3.4}$$

with $err_t^m := \mathbb{I}[Y_t^{true} \notin C_{\alpha_t^m}^m]$ equals 1 if the predicted set does not contain the true label Y_t^{true}, and 0 otherwise. According to the updating rule outlined in equation (3.3), the adjustment of α_t^m at each time t is governed by the $\nabla_{\alpha_t^m} L(\bar{\alpha}_t^m, \alpha_t^m)$, as detailed in equation (3.4). When $err_t^m = 1$, it signals that the coverage probability $1-\alpha_t^m$ is too small, resulting in a prediction set that can not encompass Y_t^{true}. Consequently, there's a necessity to enlarge the coverage probability, effectively achieved by reducing α_t^m, which would be facilitated by (3.3); given

that the denominator in the second term is always positive and the gradient will be positive in this scenario. On the other hand, when $\bar{\alpha}_t^m > \alpha_t^m$, $1 - \alpha_t^m$ leads to a prediction set that covers Y_t^{true} but also includes unnecessary labels $\mathcal{Y}' := \{Y' \in \mathcal{Y} \mid \hat{q}^m_{\bar{\alpha}_t^m} < S^m(X_t, Y') \leq \hat{q}^m_{\alpha_t^m}\}$. In such cases, optimization necessitates increasing α_t^m to avoid including unnecessary labels and output a more efficient prediction set. This adjustment is facilitated by the update rule (3.3).

Through this model-specific update process, the algorithm learns an individualized calibration level for each model. This enables better adaptation to varying prediction behaviors and improves the efficiency of the resulting conformal sets in static settings.

3.2.2 Model Selection via PMF-Based Sampling

In addition to learning a model-specific miscoverage probability, the algorithm assigns a weight w_t^m to each model $m \in [M]$, which reflects the historical performance of that model up to time t. These weights influence model selection by controlling the likelihood of using each model's associated miscoverage probability α_t^m.

After observing the true label Y_t^{true}, each model's weight is updated according to the incurred loss based on its own miscoverage probability. Specifically, the weight update rule is given by:

$$w_{t+1}^m = w_t^m \exp\left(-\epsilon L\left(\bar{\alpha}_t^m, \alpha_t^m\right)\right), \tag{3.5}$$

where $0 < \epsilon < 1$ is the learning rate. Models with lower loss (i.e., more accurate and efficient prediction sets) are rewarded with higher weights, while those with higher loss are penalized.

At each time step t, the raw weights $\{w_t^m\}_{m=1}^M$ are normalized to form a probability mass

function (PMF):

$$\bar{w}_t^m = \frac{w_t^m}{\sum_{j=1}^M w_t^j},$$

which ensures that each $\bar{w}_t^m \in [0,1]$ and $\sum_{m=1}^M \bar{w}_t^m = 1$. The normalized weight $\bar{w}_t^m$ represents the likelihood of selecting model m's miscoverage probability α_t^m.

Model selection is then performed by sampling an index $\hat{m} \in [M]$ according to the PMF $\bar{\boldsymbol{w}}_t = (\bar{w}_t^1, \ldots, \bar{w}_t^M)$. The selected model's miscoverage probability $\alpha_t^{\hat{m}}$ is used to construct the prediction set for the new input X_t. The prediction set is defined as:

$$C^{\hat{m}}_{\alpha_t^{\hat{m}}}(X_t) = \{Y \in \mathcal{Y} \mid S^{\hat{m}}(X_t, Y) \leq \hat{q}^{\hat{m}}_{\alpha_t^{\hat{m}}}\},$$

where $\hat{q}^{\hat{m}}_{\alpha_t^{\hat{m}}}$ is computed using the quantile formula in (2.1) with $\alpha = \alpha_t^{\hat{m}}$ and model $m = \hat{m}$.

After the true label Y_t^{true} is observed, all weights $\{w_t^m\}_{m=1}^M$ and miscoverage probabilities $\{\alpha_t^m\}_{m=1}^M$ are updated in parallel using (3.5) and (3.3), respectively. This complete procedure is summarized in Algorithm 1. The MOCP algorithm achieves a runtime of $\mathcal{O}(T)$ when the number of models M is constant, making it efficient for long prediction horizons.

3.3 Theoretical guarantees

In the present section, we analyze the static regret of MOCP under the assumption that the data distribution does not change over time. Given that the environment is static, there exists a miscoverage probability that can minimize the loss function for each model $m \in [M]$ over $[T]$, denoted as α^m. Let

$$\alpha^m = \arg\min_{\alpha_t^m} \sum_{t=1}^T L(\bar{\alpha}_t^m, \alpha_t^m)$$

Algorithm 1 Multi-model Ensemble Online Conformal Prediction (MOCP)

Require: $\alpha \in [0,1]$, learning rate $\eta \geq 0$, and step size $\epsilon \in (0,1)$.
  $w_1^m \leftarrow \frac{1}{M}$ {Initialization of weight for each model}
  $\alpha_1^m \leftarrow \alpha$ {Initialization of miscoverage probability for each model}
  for $t \in [T]$ **do**
    Observe $X_t \in \mathcal{X}$.
    Calculate normalized weights by $\bar{w}_t^m = \frac{w_t^m}{\sum_{j=1}^M w_t^j}$, $\forall m \in [M]$.
    Select one of the miscoverage probabilities $\{\alpha_t^m\}_{m=1}^M$ according to PMF $\bar{\boldsymbol{w}}_t = (\bar{w}_t^m)_{m=1}^M$.

    Observe true label Y_t^{true} and compute optimal value $\bar{\alpha}_t^m$ $\forall m \in [M]$ with (3.2).
    for $m \in [M]$ **do**
      Obtain loss $L(\bar{\alpha}_t^m, \alpha_t^m)$.
      Update w_{t+1}^m with (3.5).
      Update α_{t+1}^m with (3.3).
    end for
  end for

denote the optimal fixed value for model m, where the loss is defined by the pinball function introduced in (3.1). The MOCP algorithm aims to achieve performance close to the best model, which corresponding miscoverage probability can be obtained by :

$$\alpha^{m^*} = \underset{\{\alpha^m,\, m\in[M]\}}{\arg\min} \sum_{t=1}^{T} L(\bar{\alpha}_t^m, \alpha^m). \tag{3.6}$$

By adapting both model selection and calibration over time, MOCP seeks to minimize the regret relative to this optimal baseline while maintaining valid prediction set coverage. The following theorem demonstrates that MOCP achieves sublinear regret (See proof in A.1).

Theorem 3.1. *Algorithm 1 achieves the following regret bound in a static environment*

$$\sum_{t=1}^{T}\sum_{m=1}^{M} \bar{w}_t^m L(\bar{\alpha}_t^m, \alpha_t^m) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha^{m^*}) \leq \sqrt{T}\left(\frac{(1+2\eta)^2}{2\eta} + \frac{\eta}{2\alpha} + \ln M + (1+\eta)^2\right). \tag{3.7}$$

3.4 Conclusion

This chapter introduced the MOCP algorithm for constructing prediction sets in static environments. The proposed method assigns a distinct miscoverage probability to each learning model and updates it online using the pinball loss and SF-OGD. In parallel, model selection is performed through a PMF, allowing the algorithm to favor models that consistently yield efficient prediction sets. A regret analysis demonstrated that MOCP achieves sublinear regret with respect to the best fixed model and coverage level in hindsight.

While MOCP is well-suited for static environments, it does not explicitly account for temporal distribution shifts. In the next chapter, we address this limitation by extending the framework to dynamic environments through a hierarchical ensemble strategy designed to achieve strongly adaptive regret.

Chapter 4

Strongly Adaptive Multi-model Ensemble Conformal Prediction

4.1 Introduction

While the MOCP algorithm achieves valid coverage prediction sets in static environments, it does not explicitly address the challenges posed by temporal distribution shifts. In dynamic settings, the underlying data distribution may evolve over time, rendering a fixed strategy suboptimal across the entire prediction horizon. As a result, prediction sets calibrated based on earlier data may become misaligned with more recent samples.

To address this limitation, we propose the **S**trongly **A**daptive **M**ulti-model **O**nline **C**onformal **P**rediction (SAMOCP) algorithm, which extends the MOCP framework to dynamic environments. SAMOCP leverages a pool of experts, each corresponding to an instance of MOCP, and assigns to each expert a lifetime interval. This hierarchical structure enables the algorithm to track changes in the data distribution at multiple temporal resolutions.

At each time step, SAMOCP selects an expert probabilistically based on its historical performance, and uses the expert's selected model and miscoverage level to construct the prediction set. By integrating model selection and coverage calibration across multiple temporal scales, SAMOCP achieves both valid coverage and *strongly adaptive regret*—a performance guarantee that ensures sublinear regret over any interval of time, regardless of its position or length.

The remainder of this chapter is organized as follows. Section 4.2 presents the SAMOCP algorithm in detail, including the expert construction strategy and the mechanism for model selection and weight updates. Section 4.3 provides theoretical guarantees for SAMOCP, including coverage validity, strongly adaptive regret, and dynamic regret bounds.

4.2 Multi-model Conformal Prediction in Dynamic Environment

Through Algorithm 1, we demonstrated how to select the suitable miscoverage probability of a model that has achieved lower loss compared to others over previous time steps. However, this approach assumes a static environment where the underlying data distribution remains constant, which may limit its effectiveness in dynamic environments where distribution shifts occur over time.

Moreover, the performance of the algorithm is highly sensitive to the choice of the stepsize ϵ in (3.5), which controls how aggressively the algorithm updates model weights. In environments with unknown or unpredictable distribution shifts, a large ϵ enables faster adaptation to abrupt changes but may introduce instability, while a small ϵ ensures smoother updates but may lead to delayed responses to sudden shifts. Thus, the optimal setting of ϵ depends on the variability of the environment, which is typically unknown in practice.

These limitations motivate the development of a more flexible and robust framework capable of adapting across multiple timescales. Instead of relying on a single learning rate or fixed model selector, we propose a hierarchical ensemble of learners, each tailored to a different temporal resolution. This leads to the Strongly Adaptive Multi-model Online Conformal Prediction (SAMOCP) algorithm, which maintains a pool of experts with varying lifetimes and adapts both model selection and calibration to the current distribution. The following subsections explain the structure of SAMOCP in two parts. Subsection 4.2.1 presents how experts are constructed, and Subsection 4.2.2 details the mechanism each expert uses to select update weights based on performance.

4.2.1 Expert Construction and Lifetime

In dynamic environments where the data distribution may shift over time, relying on a single conformal predictor can lead to performance degradation. To address this, SAMOCP maintains a pool of experts—each corresponding to an instance of MOCP—initialized at different time steps and configured to operate over different lifetimes. Each expert has its own specific lifetime and operates independently of others. At the end of its lifetime, the expert becomes inactive, ensuring that it no longer affects the decision-making process; refer to Figure 4.1 for an illustration. This strategy prevents outdated experts—trained on older data distributions—from contributing to the selection of the suitable miscoverage probability in dynamic environments.

Subsequently, SAMOCP dynamically selects the appropriate expert for each time t and utilizes its chosen miscoverage probability to construct the prediction set. This dynamic selection mechanism enables the algorithm to track and respond to distributional changes more effectively.

At each time t, a new expert is created with a specific lifetime $\lambda(t)$. The lifetime of each

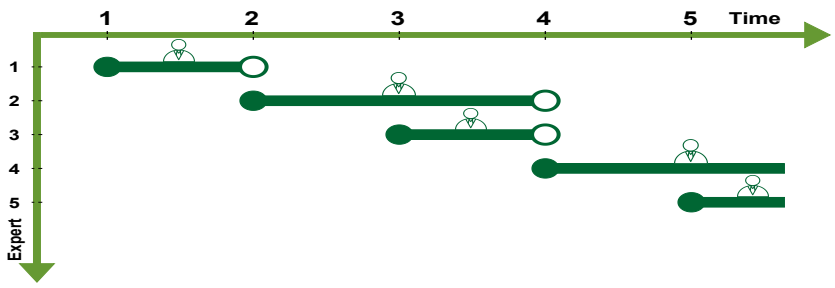


Figure 4.1: Expert creation over 5 time steps using lifetime formula (4.1) when $g = 1$.

expert is determined by the time step at which it was created and a hyperparameter $g \in \mathbb{N}$, as described in [5]:

$$\lambda(t) = g \cdot \max_{n \in \mathbb{Z}} \{2^n : t \equiv 0 \mod 2^n\}. \tag{4.1}$$

This schedule ensures that short-lived experts are created more frequently, allowing fast adaptation to recent changes, while long-lived experts persist over broader intervals, providing stability and robustness to noise or transient fluctuations.

Each expert remains active over the interval $[t, t + \lambda(t) - 1]$, after which it becomes inactive. At any given time, multiple experts may be active in parallel, each focusing on a different temporal resolution. This ensemble of experts, operating at overlapping lifetimes, allows SAMOCP to balance responsiveness and long-term consistency.

4.2.2 Expert-Level Model Selection and Weight Update

In a dynamic setting with unknown distribution shifts, the best-performing model may vary across time and underlying distributions. As a result, at each time step t, different experts may rely on different models depending on how recently they were initialized and how much they have adapted to the current data distribution. This leads to a scenario where recently created experts might better capture current dynamics, while older experts may still reflect

past conditions.

At each time t, each active expert $n \in \mathcal{A}(t)$ selects a model $\hat{m} \in [M]$ using its own internal MOCP instance. The miscoverage probability of model $\hat{m}$ selected by expert n is denoted as $\alpha_t^{\hat{m}n}$, and its optimal value—the smallest one that still ensures coverage—is denoted as $\bar{\alpha}_t^{\hat{m}n}$.

To determine which expert's prediction to use, SAMOCP assigns a weight h_t^n to each expert. These weights are updated online to reflect the predictive performance of each expert. The step size used in the weight update is scaled based on the expert's lifetime to ensure stability over longer horizons:

$$\epsilon^n := \min\left(\epsilon, \frac{\sigma}{\sqrt{\lambda(n)}}\right),$$

where $\sigma > 1$ is a constant, ϵ is a global learning rate, and $\lambda(n)$ is the lifetime of expert n as defined in (4.1). This scaling ensures that short-lived experts update more aggressively, while long-lived experts change more conservatively.

The weight update rule for expert n is defined as:

$$h_{t+1}^n = \begin{cases} \epsilon^n & \text{if } t = n-1 \\ h_t^n \exp\left(-\epsilon^n \cdot r_t^n\right) & \text{if } t \in [n, n+\lambda(n)-1) \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

where r_t^n is the relative loss of expert n compared to the selected expert $\hat{n}$, defined as

$$r_t^n = L(\bar{\alpha}_t^{\hat{m}\hat{n}}, \alpha_t^{\hat{m}\hat{n}}) - L(\bar{\alpha}_t^{\hat{m}n}, \alpha_t^{\hat{m}n}).$$

This update encourages experts with lower loss to retain higher weights. Let $\mathcal{A}(t)$ denote

the set of active experts at time t. SAMOCP selects among them probabilistically using a normalized distribution:

$$\bar{h}_t^n = \frac{h_t^n}{\sum_{i \in \mathcal{A}(t)} h_t^i},$$

forming a probability mass function $\bar{\boldsymbol{h}}_t := (\bar{h}_t^n)_{n \in \mathcal{A}(t)}$. Each expert is selected with probability proportional to its normalized weight, and the chosen expert's model and miscoverage probability are used to generate the prediction set. Algorithm 2 summarizes the full SAMOCP procedure.

Remark 4.1 (Computational Complexity)**.** *It can be observed from the lifetime schedule in* (4.1) *that the maximum number of active experts at any time t is bounded by $g\lfloor \log_2 t \rfloor$. Therefore, the total computational complexity of SAMOCP over a horizon of length T is* $\mathcal{O}(T \log T)$

Algorithm 2 Strongly Adaptive Multi-model Ensemble Online Conformal Prediction (SAMOCP)

Require: $\alpha \in [0, 1]$, hyperparameters $\eta \geq 0, \epsilon \in (0, 1)$, and $\sigma > 1$.
  for $t \in [T]$ **do**
    Create new expert n (where $n = t$) by Algorithm 1$(\alpha_{t-1}^{\hat{m}\hat{n}}, \eta, \epsilon^n)$.
    Remove experts whose lifetime has been finished.
    Every active expert selects miscoverage probability from M options.
    Calculate normalized weights by $\bar{h}_t^n = \frac{h_t^n}{\sum_{i \in \mathcal{A}(t)} h_t^i}$.
    Select one misoverage probability from active experts according to PMF $\bar{\boldsymbol{h}}_t = (\bar{h}_t^n)_{n \in \mathcal{A}(t)}$.

    Construct prediction set for X_t using selected miscoverage probability.
    Observe true label Y_t^{true}.
    for $n \in \mathcal{A}(t)$ **do**
      Obtain learner loss $L(\bar{\alpha}_t^{\hat{m}\hat{n}}, \alpha_t^{\hat{m}\hat{n}})$.
      Update every parameters assigned to each model for expert n via Algorithm 1.
      Update h_{t+1}^n with (4.2).
    end for
  end for

4.3 Theoretical Guarantees

This section establishes the theoretical performance guarantees of SAMOCP. Specifically, we provide bounds on (i) the coverage error of the algorithm and (ii) the regret with respect to the best model and miscoverage probability under both static and dynamic settings.

Let $CovE(T) := \left|\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[err_t] - \alpha\right|$ represent the coverage error. In a dynamic setting where multiple experts are incorporated, each including M miscoverage probabilities, the expected error is calculated as $\mathbb{E}[err_t] = \sum_{n=1}^{t}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}err_t^{mn}$, where mn represents the mth model by expert n. Using the two following theorems, we prove that SAMOCP has bounded coverage error and achieves strongly adaptive regret across any time interval of arbitrary width (Proofs can be found A.2 and A.3).

Theorem 4.1. *For any $T \geq 1$ and any $\gamma \in \left(\frac{1}{2}, 1\right)$, Algorithm 2 achieves the coverage error bound*

$$CovE(T) \leq \mathcal{O}\left(\inf_{\gamma}\left\{T^{\frac{1}{2}-\gamma} + T^{\gamma-1}\beta_\gamma(T)\right\}\right), \tag{4.3}$$

where $\beta_\gamma(T)$ measures the smoothness of model weights within experts and the cumulative gradient norm for each model within experts. The definition of $\beta_\gamma(T)$ is provided in detail in equation (A.14) *in the Appendix A.2. If there exists a $\gamma \in \left(\frac{1}{2}, 1\right)$ such that $\beta_\gamma(T) \leq \tilde{\mathcal{O}}(T^\theta)$ where $\theta < 1-\gamma$, then the coverage bound* (4.3) *will be $CovE(T) \leq \tilde{\mathcal{O}}\left(T^{-\min\left(\frac{1}{2}-\gamma, \gamma-1+\theta\right)}\right) = \mathbf{o}_T(1)$.*

Theorem 4.2. *Algorithm 2 achieves strongly adaptive regret over any interval $I \subseteq [T]$ and positive constants A, B, as follows*

$$\sum_{t\in I}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) - \sum_{t\in I}L(\bar{\alpha}_t^{m^*n^*}, \alpha^{m^*n^*}) \leq A\sqrt{|I|} + B\ln T\sqrt{|I|}, \tag{4.4}$$

where

$$\alpha^{m^*n^*} = \arg\min_{\alpha^{m^*n}} \sum_{t\in I} L(\bar{\alpha}_t^{m^*n}, \alpha^{m^*n}). \tag{4.5}$$

Note that in equation (4.5)*,* α^{m^*n} *represents the miscoverage probability assigned to the best model for expert* n*, as obtained by equation* (3.6)*.*

The miscoverage probability $\alpha^{m^*n^*}$ in (4.5), is related to specific interval I, which can vary across different intervals with distinct distributions in a dynamic environment. In such settings, there is no fixed miscoverage probability $\alpha^{m^*n^*}$ that can be consistently applied over various time intervals. This necessitates establishing that SAMOCP has bounded regret in dynamic environments with respect to the time-varying benchmark in (4.8), as demonstrated by the following lemma. The proof is provided in A.4.

Lemma 4.1. *By defining the variation of the loss function to be*

$$V(L(\cdot)_{t=1}^T) := \sum_{t=1}^{T} \max_{\{m\in[M], n\in\mathcal{A}(t)\}} \left| L(\bar{\alpha}_{t+1}^{mn}, \alpha_{t+1}^{mn}) - L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) \right|. \tag{4.6}$$

We establish the following bound for the dynamic regret of Algorithm 2

$$\sum_{t=1}^{T} \sum_{n\in\mathcal{A}(t)} \sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*n^*}, \alpha_t^{m^*n^*}) \leq \tilde{\mathcal{O}}(T^{\frac{2}{3}} V^{\frac{1}{3}}(L(.)_{t=1}^T)) \tag{4.7}$$

where $\tilde{\mathcal{O}}$ *suppresses positive constants and polylogarithmic factors, e.g.,* $\log T$*. Also the best miscoverage probability at each time* t *can be obtained by*

$$\alpha_t^{m^*n^*} = \arg\min_{\{m\in[M], n\in\mathcal{A}(t)\}} L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}). \tag{4.8}$$

Lemma 4.1 establishes that the dynamic regret of SAMOCP (4.7) depends on the variation of the loss functions (4.6). In addition, it can be obtained from (4.7) that SAMCOP achieves

sublinear regret if the variation of the loss function is also sublinear, i.e., $V(L(.)_{t=1}^{T}) = \mathbf{o}(T)$.

4.4 Conclusion

This chapter introduced the SAMOCP algorithm for constructing valid and efficient prediction sets in dynamic environments. To address the limitations of single-model predictors under distribution shift, SAMOCP maintains a hierarchy of experts—each an instance of MOCP—initialized at different time steps with lifetimes governed by a doubling schedule. This hierarchical design enables the algorithm to adapt simultaneously across multiple temporal scales.

Each expert operates independently, updating its own model weights and miscoverage probabilities based on recent observations. A probabilistic selection mechanism based on expert performance is used to determine which expert's model and prediction set to use at each time step. Theoretical guarantees were established for the proposed method, including bounds on coverage error and both strongly adaptive and dynamic regret.

By integrating model selection, calibration, and expert scheduling into a unified framework, SAMOCP provides a solution for uncertainty quantification in non-stationary settings. In the next chapter, we empirically evaluate the performance of SAMOCP on classification tasks subject to distributional shifts and compare it against baseline methods.

Chapter 5

Experiments

In this chapter, the performance of the proposed method, SAMOCP, is assessed within the context of classification tasks. We conduct a comprehensive comparison with recently proposed methods in online conformal prediction for dynamic environments within classification tasks. Note that throughout the experiments in this chapter, the desired miscoverage probability α is 0.1. All experiments were performed on a workstation with NVIDIA RTX A4000 GPU.

5.1 Experimental setup

Dataset: We utilize corrupted versions of CIFAR-10, CIFAR-100 [22], and TinyImageNet [23] known as CIFAR-10C, CIFAR-100C, and TinyImageNet-C [18]. These datasets consist of 15 generated corruptions spanning 5 distinct levels of severity. The evaluation encompasses two settings: sudden and gradual distribution shifts. For both settings, the data sequence is split into batches of 500 data samples each. The severity of corruption changes (increases or decreases) after each batch of data. In the sudden shifts, the severity level alternates

between the version of the data without any corruption (severity level 0) and the most severe corruption (severity level 5). In the gradual setting, severity starts at level 0 and increases one by one after each batch until it reaches level 5. After reaching level 5, the severity decreases one by one and goes back to level 0 in subsequent batches. This cycle of increasing and decreasing severity continues throughout the experiment. Also, additional experiments on TinyImageNet-C [18] and synthetic data are provided in the Appendix, Section.

Baselines and experimental settings: We employ ResNet-50, ResNet-18 [17], GoogLeNet [34], and DenseNet-121 [19] as candidate learning models. Each active expert consists of all these learning models and needs to select the appropriate model during its active interval. The proposed method is compared with the most recent adaptive conformal prediction algorithms designed for dynamic environments, including FACI [14], ScaleFreeOGD [5], and SAOCP [5]. FACI employs a fixed number of active experts over all time steps, with each expert assigned one of the candidate learning rates for updating the miscoverage probability. ScaleFreeOGD reduces the learning rate based on the cumulative norms of gradients [27]. SAOCP allows each expert to have its own active interval, within which it operates similarly to ScaleFreeOGD. In order to show how SAMOCP results in more efficient sets compared to SAOCP in a multi-model setting, we developed a multi-model ensemble version of SAOCP, denoted as SAOCP(MM), where each expert consists of M update rules, each corresponding to a different learning model. This approach follows our multi-model approach but employs a similar rule to SAOCP for updating weights. To determine the value of g, we employed a grid search approach within the candidates $\{4, 8, 16, 24, 32, 48, 64\}$. The one that led to the smallest prediction set size (Avg Width) while maintaining reasonable coverage and runtime was selected, which was $g = 8$. While the hyperparameter g is set to 8 for both SAMOCP and SAOCP(MM), it is set to 32 for SAOCP, as in [5]. Since 4 learning models are incorporated in this section, the maximum number of updates at each time t in SAMOCP, $Mg\lfloor \log_2 t \rfloor$, is equal to that in SAOCP and SAOCP(MM), which is $32\lfloor \log_2 t \rfloor$. Meanwhile, note that randomness might be undesirable in practice, the predicted miscoverage in SAMOCP is cal-

culated in a deterministic fashion, i.e., $\alpha_t = \sum_{n \in \mathcal{A}(t)} \sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} \alpha_t^{mn}$. For every experiment conducted on the synthetic dataset, CIFAR-10C, CIFAR-100C, parameters ϵ, σ, and η were selected through grid search, with values of 0.9, 140, and 0.05, respectively.

Score Functions: We utilized the nonconformity score defined as in [1] to construct prediction sets. Let

$$S^m(X, Y) = \xi \sqrt{\max([k_Y - k_{reg}], 0)} + U_t \hat{f}_Y^m(X) + \rho(X, Y), \tag{5.1}$$

where $\hat{f}_Y^m(X)$ denotes the probability of predicting label Y for input X by model m, and U_t is a random variable sampled from a uniform distribution over the interval $[0, 1]$. The term $k_Y := |\{Y' \in \mathcal{Y} \mid \hat{f}_{Y'}^m(X) \geq \hat{f}_Y^m(X)\}|$ denotes the number of labels that have a higher or equal predicted probability than label Y according to the model's output probability distribution, e.g., the softmax output. $\rho(X, Y) := \sum_{Y'=1}^{K} \hat{f}_{Y'}^m(X) \mathbb{I}[\hat{f}_{Y'}^m(X) > \hat{f}_Y^m(X)]$ sums up the probabilities of all labels that have a higher predicted probability than label Y. The hyperparameters ξ and k_{reg} are set to 0.02 and 5 for CIFAR-100C, and 0.1 and 1 for Cifar-10C, respectively.

Evaluation Metrics: Coverage measures the percentage of instances where the true label is included in the prediction sets outputted by the conformal prediction algorithm over the period $[T]$. Avg Width refers to the average size of these prediction sets. Adaptive regret is calculated for time intervals of length 100. The metric Avg Regret represents the average of these regret values across the entire time horizon $[T]$. Run Time indicates the time required to complete each iteration of the algorithm. Lastly, Single Width measures the probability that prediction sets contain exactly one element while accurately covering the true label, highlighting cases that are most informative for predictions.

5.2 Experimental Results

5.2.1 Experiments on Real Data

Table 5.1: Results on the CIFAR-100C dataset with a gradual distribution shift.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time	Single Width
	SAMOCP	88.16 ± 0.18	**5.43 ± 0.28**	**0.92 ± 0.07**	34.87 ± 0.67	**0.29 ± 0.01**
	SAOCP(MM)	85.89 ± 0.84	6.61 ± 0.26	4.42 ± 0.51	47.80 ± 0.22	0.27 ± 0.01
DenseNet-121	FACI	89.64 ± 0.28	5.77 ± 0.62	1.18 ± 0.74	8.22 ± 0.07	0.28 ± 0.01
	ScaleFreeOGD	**89.97 ± 0.02**	6.04 ± 0.10	1.55 ± 0.06	**3.02 ± 0.06**	0.26 ± 0.01
	SAOCP	88.90 ± 0.10	5.80 ± 0.08	2.04 ± 0.04	40.74 ± 0.31	0.27 ± 0.00
ResNet-50	FACI	89.67 ± 0.33	6.50 ± 0.57	1.24 ± 0.90	8.45 ± 0.09	0.26 ± 0.01
	ScaleFreeOGD	**89.97 ± 0.02**	6.72 ± 0.13	1.58 ± 0.07	3.12 ± 0.04	0.25 ± 0.01
	SAOCP	88.79 ± 0.14	6.48 ± 0.13	2.08 ± 0.10	41.35 ± 0.34	0.25 ± 0.00
ResNet-18	FACI	89.56 ± 0.28	6.82 ± 0.77	1.17 ± 0.75	8.39 ± 0.06	0.25 ± 0.01
	ScaleFreeOGD	89.96 ± 0.02	7.29 ± 0.19	1.55 ± 0.07	3.05 ± 0.04	0.23 ± 0.01
	SAOCP	88.76 ± 0.22	6.9 ± 0.17	2.06 ± 0.07	41.23 ± 0.26	0.24 ± 0.01
GoogLeNet	FACI	89.63 ± 0.30	6.33 ± 0.74	1.10 ± 0.78	8.30 ± 0.09	0.27 ± 0.01
	ScaleFreeOGD	89.96 ± 0.01	6.71 ± 0.15	1.52 ± 0.06	3.04 ± 0.04	0.24 ± 0.00
	SAOCP	88.68 ± 0.11	6.38 ± 0.12	2.07 ± 0.07	41.13 ± 0.39	0.26 ± 0.01

Table 5.1 presents the performance of SAMOCP for the classification task on the CIFAR-100C dataset under a gradual shift setting, where each method receives $8,550$ data points sequentially. It is evident that the performance of previous methods, particularly in terms of prediction set size and single-width prediction sets, depends on the learning model employed. The proposed method SAMOCP outperforms existing methods by creating smaller prediction sets, lower regret, and more single-width prediction sets that correctly cover the true label, while also achieving coverage close to the targeted level. It is noteworthy that SAMOCP surpasses every variant of previous methods in these aspects. Furthermore, SAMOCP is faster than SAOCP and SAOCP(MM), despite having the same maximum number of updates at each time t.

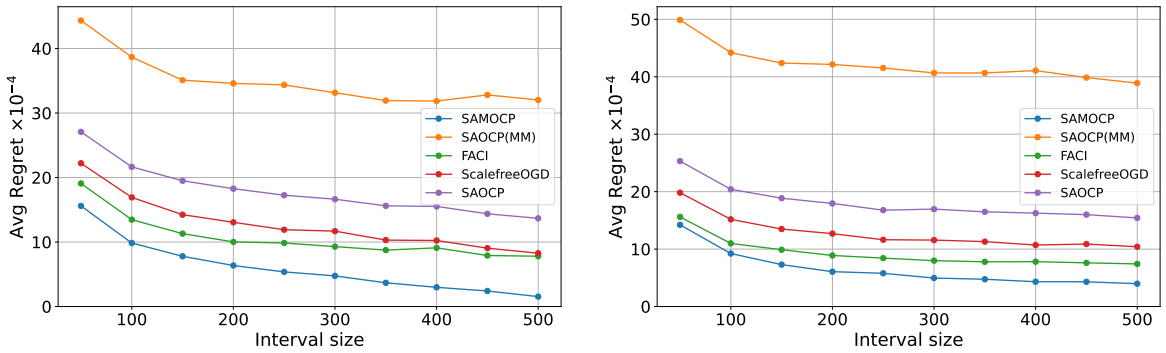
In dynamic environments, data distribution does not necessarily shift gradually. Instead, we may encounter abrupt distribution shifts, with significant differences between data distributions in two successive time slots. To demonstrate how SAMOCP behaves in such

Table 5.2: Results on the CIFAR-10C dataset with a sudden distribution shift.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time	Single Width
	SAMOCP	88.37 ± 0.23	**1.24 ± 0.06**	**0.98 ± 0.11**	33.75 ± 0.34	**0.69 ± 0.03**
	SAOCP(MM)	86.80 ± 2.39	1.45 ± 0.13	3.87 ± 1.05	47.08 ± 0.19	0.56 ± 0.05
DenseNet-121	FACI	89.57 ± 0.37	1.30 ± 0.12	1.46 ± 0.73	8.11 ± 0.10	0.68 ± 0.05
	ScaleFreeOGD	**89.99 ± 0.01**	1.46 ± 0.02	1.71 ± 0.04	2.92 ± 0.07	0.52 ± 0.02
	SAOCP	88.77 ± 0.18	1.41 ± 0.02	2.24 ± 0.06	39.62 ± 0.22	0.54 ± 0.01
ResNet-50	FACI	89.74 ± 0.35	1.50 ± 0.04	1.35 ± 0.93	8.11 ± 0.08	0.55 ± 0.01
	ScaleFreeOGD	89.98 ± 0.01	1.52 ± 0.01	1.71 ± 0.05	**2.89 ± 0.04**	0.54 ± 0.01
	SAOCP	89.12 ± 0.08	1.51 ± 0.01	2.17 ± 0.06	40.25 ± 0.27	0.53 ± 0.01
ResNet-18	FACI	89.63 ± 0.34	1.36 ± 0.13	1.52 ± 0.76	8.11 ± 0.08	0.64 ± 0.05
	ScaleFreeOGD	**89.99 ± 0.01**	1.52 ± 0.02	1.69 ± 0.06	2.91 ± 0.06	0.49 ± 0.01
	SAOCP	88.83 ± 0.06	1.48 ± 0.02	2.24 ± 0.07	40.16 ± 0.22	0.51 ± 0.01
GoogLeNet	FACI	89.73 ± 0.34	1.43 ± 0.06	1.41 ± 0.89	8.10 ± 0.10	0.58 ± 0.02
	ScaleFreeOGD	89.99 ± 0.02	1.46 ± 0.01	1.70 ± 0.06	**2.89 ± 0.04**	0.55 ± 0.00
	SAOCP	89.09 ± 0.14	1.44 ± 0.01	2.17 ± 0.08	40.13 ± 0.17	0.55 ± 0.01

environments, another experiment was conducted, in which SAMOCP can successfully track these sharp transitions and select a suitable learning model for creating a prediction set. Experimental results on CIFAR-10C are detailed in Table 5.2, where the proposed algorithm again outperforms previous methods in terms of prediction set size, regret, and single width prediction sets that accurately cover the true labels while maintaining coverage close to the target value.

To demonstrate that SAMOCP achieves the lowest regret over different intervals compared to existing methods, we illustrate the regret for various interval sizes in Figure 5.1. For each existing method, there are 4 different regrets corresponding to the 4 learning models used and the lowest regret is depicted. The results show that our method consistently leads to lower regret than the best version of each previous method across different learning models. Note that a lower regret implies that the algorithm adapts faster to changes. The regret calculated over different time intervals indicates the algorithm's adaptivity in capturing the distribution shift at different time scales. Therefore, Figure 5.1 indicates that the SAMOCP can adapt faster to distribution shifts compared to benchmarks in various time scales. Furthermore, we include experiments on synthetic data and another real dataset, TinyImageNet-C, using different sets of learning models that do not necessarily contain 4 models. This demonstrates



(a) CIFAR-10C (b) CIFAR-100C

Figure 5.1: Evaluation of average regret over different interval sizes $(50, 100, \ldots, 500)$.

how SAMOCP can rely on a mixture of learning models over the period $[T]$ and select the appropriate one for each distribution setting.

SAMOCP vs. SAOCP (equal lifetimes): We compared our method with a specific version of SAOCP where the hyperparameter g in equation (4.1) is set to 8, giving both SAOCP and SAMOCP experts the same lifetime. Our results are demonstrated in Table 5.3 for CIFAR-10C and in Table 5.4 for CIFAR-100C. For both cases, we observe the advantage of utilizing multiple learning models instead of a single model, as our proposed method, SAMOCP, significantly outperforms SAOCP across all four learning models. SAMOCP achieved better results in terms of coverage, average width, average regret, and single width compared to SAOCP.

Table 5.3: Performance evaluation of SAOCP with same lifetime time as SAMOCP for sudden distribution shift setting on CIFAR-10C.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time	Single Width
	SAMOCP	**88.37 ± 0.23**	**1.24 ± 0.06**	**0.98 ± 0.11**	33.75 ± 0.34	**0.69 ± 0.03**
DenseNet-121	SAOCP	87.16 ± 0.10	1.45 ± 0.03	4.36 ± 0.14	**14.73 ± 0.09**	0.53 ± 0.01
ResNet-50	SAOCP	87.87 ± 0.15	1.52 ± 0.01	3.91 ± 0.11	14.78 ± 0.07	0.52 ± 0.01
ResNet-18	SAOCP	87.16 ± 0.18	1.51 ± 0.02	4.38 ± 0.16	14.85 ± 0.08	0.51 ± 0.01
GoogLeNet	SAOCP	87.74 ± 0.10	1.45 ± 0.01	3.92 ± 0.15	14.84 ± 0.06	0.54 ± 0.01

SAMOCP vs. MOCP: To validate the advantage of SAMOCP versus MOCP in dynamic environments, we conducted experiments for both sudden and gradual distribution shifts.

Table 5.4: Performance evaluation of SAOCP with same lifetime time as SAMOCP for gradual distribution shift setting on CIFAR-100C.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time	Single Width
	SAMOCP	**88.16 ± 0.18**	**5.43 ± 0.28**	**0.92 ± 0.07**	34.87 ± 0.67	**0.29 ± 0.01**
DenseNet-121	SAOCP	87.33 ± 0.14	6.12 ± 0.16	4.03 ± 0.06	**14.89 ± 0.06**	0.27 ± 0.00
ResNet-50	SAOCP	87.22 ± 0.12	6.75 ± 0.16	3.99 ± 0.14	15.01 ± 0.07	0.26 ± 0.00
ResNet-18	SAOCP	87.15 ± 0.14	7.24 ± 0.22	4.10 ± 0.11	15.02 ± 0.08	0.25 ± 0.00
GoogLeNet	SAOCP	87.09 ± 0.16	6.78 ± 0.12	4.14 ± 0.13	14.98 ± 0.09	0.26 ± 0.00

see Tables 5.5 and 5.6. In both tables, SAMOCP outperforms MOCP in terms of Average Width and Single Width metrics.

Table 5.5: Comparison of MOCP and SAMOCP on the CIFAR-10C dataset with a sudden distribution shift.

Method	Coverage (%)	Avg Width	Single Width
MOCP	**89.96 ± 0.33**	1.29 ± 0.08	0.67 ± 0.04
SAMOCP	88.37 ± 0.23	**1.24 ± 0.06**	**0.69 ± 0.03**

Table 5.6: Comparison of MOCP and SAMOCP on the CIFAR-100C dataset with a gradual distribution shift.

Method	Coverage (%)	Avg Width	Single Width
MOCP	**89.77 ± 0.19**	5.85 ± 0.28	0.28 ± 0.01
SAMOCP	88.16 ± 0.18	**5.43 ± 0.28**	**0.29 ± 0.01**

TinyImageNet Dataset: We conduct experiments on TinyImageNet dataset, which features 200 distinct classes, under a gradual distribution shift. We have also incorporated a new mixture of learning models—GoogLeNet, DenseNet-121, EfficientNet-B0, and MobileNet-V2 [25] to demonstrate the performance of our algorithm. In Table 5.7, we detail our proposed method's comparison with previous methods, where, once again, our algorithm is able to achieve smaller prediction set sizes while maintaining coverage close to the target value of $1-\alpha$. It is reasonable to study prediction sets in situations where we achieve coverage close to the desired level. The SAOCP(MM) method achieves coverage of less than 85%, which is significantly below the target value. Therefore, we do not consider its prediction sets in our comparison. As demonstrated in the table, SAMOCP obtains smaller prediction sets and operates faster than SAOCP. For the TinyImageNet-C dataset, the parameters η, ξ and

k_{reg} are set to 0.025, 0.01 and 20, respectively. Other parameters remain consistent with previous experiments. All real datasets are downloaded from the Zenodo repository.

Table 5.7: Results on the TinyImageNet-C dataset with a gradual distribution shift.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time	Single Width
	SAMOCP	87.73 ± 0.35	**171.64 ± 1.34**	1.28 ± 0.20	35.22 ± 0.72	0
	SAOCP(MM)	84.91 ± 1.22	165.93 ± 3.32	4.80 ± 0.77	48.55 ± 0.15	0
GoogLeNet	FACI	89.69 ± 0.29	177.93 ± 1.21	1.07 ± 0.67	8.48 ± 0.06	0
	ScaleFreeOGD	**89.96 ± 0.01**	178.38 ± 0.49	1.41 ± 0.08	**3.16 ± 0.03**	0
	SAOCP	88.36 ± 0.07	174.81 ± 0.50	2.02 ± 0.08	40.83 ± 0.32	0
DenseNet-121	FACI	89.68 ± 0.31	176.66 ± 1.10	1.30 ± 0.74	8.54 ± 0.11	0
	ScaleFreeOGD	89.95 ± 0.02	177.24 ± 0.64	1.53 ± 0.06	3.18 ± 0.05	0
	SAOCP	88.48 ± 0.12	173.79 ± 0.57	2.09 ± 0.04	40.09 ± 0.36	0
Efficientnet_b0	FACI	89.64 ± 0.35	176.78 ± 1.03	1.01 ± 0.61	8.56 ± 0.07	0
	ScaleFreeOGD	89.95 ± 0.01	177.23 ± 0.63	1.37 ± 0.06	3.17 ± 0.05	0
	SAOCP	88.30 ± 0.18	173.52 ± 0.63	1.98 ± 0.08	41.06 ± 0.15	0
Mobilenet_v2	FACI	89.69 ± 0.30	175.26 ± 0.95	**0.98 ± 0.52**	8.55 ± 0.07	0
	ScaleFreeOGD	89.95 ± 0.01	175.82 ± 0.51	1.36 ± 0.07	3.19 ± 0.04	0
	SAOCP	88.30 ± 0.15	171.83 ± 0.55	1.94 ± 0.06	40.60 ± 0.23	0

5.2.2 Experiments on Synthetic Data

Additional experiments have been done using synthetic data to compare our proposed method, SAMOCP, with recent adaptive conformal prediction methods designed for dynamic settings. We analyze our experiments with a new set of learning models to demonstrate how our proposed method can effectively utilize a mixture of learning models to cope with distribution shifts in a dynamic environment with unknown distribution changes. Additionally, experiments with synthetic data also confirm that SAMOCP maintains its advantages when varying the number of learning models. Specifically, we conducted experiments for cases with 2 and 3 learning models.

Data Generation: To generate synthetic data that mimics real-world scenarios, we employ two distinct transformation sequences. The first transformation sequence introduces visual noise and blur effects through the application of Gaussian blur and random Gaussian noise. This approach aims to subtly degrade image clarity, simulating real-life challenges

such as camera focus issues or atmospheric conditions like fog or mist. The Gaussian blur is applied with moderate settings, while random noise is incorporated to simulate sensor noise or digital compression artifacts commonly encountered in digital imagery. The second transformation sequence focuses on color manipulation. By adjusting image attributes such as brightness, contrast, saturation, and hue in minor increments, we challenge the models to perform reliably under varying lighting conditions and color settings—typical variations that occur due to different times of the day or inconsistencies in camera settings. Additionally, a random conversion of some images to grayscale is employed to further challenge the models' dependency on color information.

In this experiment, two distinct datasets each containing 3000 images are generated from each transformation type. These datasets are designed with a fixed number of 20 classes and hyperparameters ξ and k_{reg} are set to 0.1 and 4, respectively. The variations between the datasets are due to random elements introduced during image processing, such as differences in which pixels are affected by noise or how color properties are altered. This randomness ensures each dataset contains unique instances, even though they stem from the same transformation principles. By concatenating images from the two datasets, the experiment simulates both gradual and sudden distribution shifts. Gradual shifts are seen within the datasets from a single transformation, while sudden shifts occur when switching between datasets from different transformations.

Table 5.8: Results on the generated synthetic dataset for Efficientnet_b0 and GoogLeNet learning models.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time
	SAMOCP	87.90 ± 0.26	**17.54 ± 0.01**	0.48 ± 0.36	19.51 ± 0.03
	SAOCP(MM)	81.98 ± 7.35	16.54 ± 1.45	6.97 ± 2.59	26.30 ± 0.24
Efficientnet_b0	FACI	89.80 ± 0.31	17.96 ± 0.05	**0.30 ± 0.18**	6.82 ± 0.02
	ScaleFreeOGD	**89.96 ± 0.00**	18.05 ± 0.01	1.61 ± 0.01	**2.45 ± 0.00**
	SAOCP	88.73 ± 0.08	17.82 ± 0.01	2.18 ± 0.02	34.43 ± 0.06
GoogLeNet	FACI	89.66 ± 0.30	18.05 ± 0.09	1.36 ± 0.79	6.89 ± 0.02
	ScaleFreeOGD	89.95 ± 0.00	18.07 ± 0.03	1.82 ± 0.03	2.46 ± 0.00
	SAOCP	88.39 ± 0.07	17.75 ± 0.02	2.55 ± 0.03	34.78 ± 0.06

We conducted experiments using a distinct set of learning models. As shown in Table 5.8, we

Table 5.9: Results on the generated synthetic dataset for EfficientNet-B0, DenseNet-121, and GoogLeNet learning models.

Model	Method	Coverage (%)	Avg Width	Avg Regret($\times 10^{-3}$)	Run Time
	SAMOCP	88.04 ± 0.31	**17.60 ± 0.04**	0.65 ± 0.45	26.39 ± 0.37
	SAOCP(MM)	80.95 ± 7.75	16.29 ± 1.55	7.06 ± 3.03	34.80 ± 0.36
Efficientnet_b0	FACI	89.80 ± 0.31	17.96 ± 0.05	**0.30 ± 0.18**	7.37 ± 0.07
	ScaleFreeOGD	**89.96 ± 0.00**	18.05 ± 0.01	1.61 ± 0.01	**2.65 ± 0.01**
	SAOCP	88.73 ± 0.08	17.82 ± 0.01	2.18 ± 0.02	37.25 ± 0.15
DenseNet-121	FACI	89.72 ± 0.32	18.04 ± 0.07	1.07 ± 0.73	7.42 ± 0.05
	ScaleFreeOGD	89.95 ± 0.01	18.06 ± 0.02	1.81 ± 0.04	2.65 ± 0.03
	SAOCP	88.39 ± 0.09	17.72 ± 0.01	2.53 ± 0.02	37.83 ± 0.16
GoogLeNet	FACI	89.66 ± 0.30	18.05 ± 0.09	1.36 ± 0.79	7.49 ± 0.06
	ScaleFreeOGD	89.95 ± 0.00	18.07 ± 0.03	1.82 ± 0.03	2.66 ± 0.03
	SAOCP	88.39 ± 0.07	17.75 ± 0.02	2.55 ± 0.03	37.90 ± 0.21

incorporated two learning models, Efficientnet_b0 and GoogLeNet, where we achieved the smallest prediction set size with coverage close to the target. The SAOCP for GoogLeNet obtained an average width close to our method; however, it is noteworthy that the maximum number of updates in SAOCP is twice that of SAMOCP, demonstrating that our algorithm achieved this result with lower computational costs. Additionally, its regret is almost five times larger than our method's. We also provide synthetic data analysis using another set of learning models consisting of GoogLeNet, DenseNet-121, and EfficientNet-B0 [35], as detailed in Table 5.9 where our method again was able to construct smaller prediction sets. It should also be noted that, due to severely corrupted data in our synthetic dataset, none of the models were able to produce single width prediction sets that cover true labels. Note that since a coverage level below 85% is significantly lower than the desired target of 90%, such values are considered unacceptable and are marked with an underline in the tables.

Chapter 6

Conclusion

6.1 Conclusion

This thesis addressed the problem of conformal prediction in dynamic environments, where data distributions may shift unpredictably over time. Recognizing that relying on a single fixed model limits efficiency of conformal predictors, we proposed a novel framework that leverages multiple models in an online setting.

We first introduced MOCP, a multi-model conformal prediction algorithm tailored for static environments. MOCP dynamically selects among candidate models based on their historical predictive performance and updates model-specific miscoverage probabilities using online optimization. Building on this foundation, we developed SAMOCP—a strongly adaptive extension that operates effectively under distribution shifts. SAMOCP maintains a hierarchy of experts, each representing a MOCP instance with a different lifetime, allowing it to adapt to changes in data distribution at multiple temporal scales.

Theoretically, we established that SAMOCP achieves strongly adaptive regret across any

arbitrary time interval and provides valid marginal coverage. Empirically, we evaluated the algorithm on both real-world and synthetic datasets exhibiting sudden and gradual shifts. Results demonstrate that SAMOCP consistently produces more informative prediction sets, achieves near-target coverage, and outperforms existing adaptive conformal prediction methods, even when they use their best-performing baseline models.

Together, these contributions offer a principled and practical framework for uncertainty quantification in non-stationary settings, paving the way for more robust decision-making in real-time learning systems.

6.2 Future Work

One important direction for future research lies in improving the scalability and efficiency of multimodel online conformal prediction algorithms. In practice, the model pool may contain learning models with consistently poor performance, which can inflate the size of the constructed prediction sets and reduce their utility. Moreover, maintaining a large number of models introduces considerable computational overhead, as parameters—such as adaptive miscoverage probabilities—must be updated for each model individually at every iteration. To address these limitations, future algorithms could explore strategies for selective model activation, where only a subset of effective models is engaged at each time step. This approach has the potential to reduce computational cost while preserving coverage guarantees and improving prediction set tightness.

Bibliography

[1] A. Angelopoulos, S. Bates, M. Jordan, and J. Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2020.

[2] V. Balasubramanian, S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations, and Applications*. Newnes, 2014.

[3] R. Barber, E. Candès, A. Ramdas, and R. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

[4] O. Besbes, Y. Gur, and A. Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[5] A. Bhatnagar, H. Wang, C. Xiong, and Y. Bai. Improved online conformal prediction via strongly adaptive online learning. In *International Conference on Machine Learning*, pages 2337–2363. PMLR, 2023.

[6] H. Boström, H. Linusson, T. Löfström, and U. Johansson. Accelerating difficulty estimation for conformal regression forests. *Annals of Mathematics and Artificial Intelligence*, 81:125–144, 2017.

[7] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R. Schapire, and M. Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

[8] A. Daniely, A. Gonen, and S. Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411. PMLR, 2015.

[9] T. Ding, A. Angelopoulos, S. Bates, M. Jordan, and R. Tibshirani. Class-conditional conformal prediction with many classes. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[10] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159, 2011.

[11] S. Ghiasvand, M. Alizadeh, and R. Pedarsani. Decentralized low-rank fine-tuning of large language models. *arXiv preprint arXiv:2501.15361*, 2025.

[12] S. Ghiasvand, A. Reisizadeh, M. Alizadeh, and R. Pedarsani. Robust decentralized learning with local updates and gradient tracking. *IEEE Transactions on Networking*, 2025.

[13] I. Gibbs and E. Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

[14] I. Gibbs and E. Candès. Conformal inference for online prediction with arbitrary distribution shifts. *arXiv preprint arXiv:2208.08401*, 2022.

[15] E. Hajihashemi and Y. Shen. Multi-model ensemble conformal prediction in dynamic environments. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[16] E. Hazan, A. Rakhlin, and P. Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[20] H. N. Jazi, B. Sun, R. Boutaba, and X. Tan. Posted price mechanisms for online allocation with diseconomies of scale. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2710–2728, New York, NY, USA, 2025. Association for Computing Machinery.

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

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[23] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. Technical Report 7, CS 231N, 2015.

[24] A. Levy, M. Agrawal, A. Satyanarayan, and D. Sontag. Assessing the impact of automated suggestions on decision-making: Domain experts mediate model errors but take less initiative. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[25] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2021.

[26] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[27] F. Orabona and D. Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.

[28] H. Papadopoulos, V. Vovk, and A. Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40:815–840, 2011.

[29] Y. Romano, E. Patterson, and E. Candes. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[30] Y. Romano, M. Sesia, and E. Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

[31] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[32] F. Shi, C. Ong, and C. Leckie. Applications of class-conditional conformal predictor in multi-class classification. In *2013 12th International Conference on Machine Learning and Applications*, volume 1, pages 235–239. IEEE, 2013.

[33] E. Straitouri, L. Wang, N. Okati, and M. Rodriguez. Improving expert predictions with conformal prediction. In *International Conference on Machine Learning*, pages 32633–32653. PMLR, 2023.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[35] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[36] R. Tibshirani, R. Foygel Barber, E. Candès, and A. Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[37] V. Vovk. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60, 1995.

[38] V. Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.

[39] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*, volume 29. Springer, New York, 2005.

[40] M. Zaffran, O. Féron, Y. Goude, J. Josse, and A. Dieuleveut. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pages 25834–25866. PMLR, 2022.

Appendix A

A.1 Proof of Theorem 3.1

Algorithm 1 has the following regret bound

$$\sum_{t=1}^{T}\sum_{m=1}^{M} \bar{w}_t^m L(\bar{\alpha}_t^m, \alpha_t^m) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha^{m^*}) \leq \sqrt{T}\left(\frac{(1+2\eta)^2}{2\eta} + \frac{\eta}{2\alpha} + \ln M + (1+\eta)^2\right),$$

Where α^{m^*} can be obtained via (3.6). To prove the theorem, we first introduce and prove the following two lemmas

Lemma A.1. *for miscoverage probability assigned to any model $\tilde{m} \in [M]$, we have the following bound*

$$\sum_{t=1}^{T} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{\tilde{m}}, \alpha^{\tilde{m}}) \leq \frac{\sqrt{T}}{2\eta}(1+2\eta)^2 + \frac{\eta\sqrt{T}}{2\alpha},$$

where $\alpha^{\tilde{m}} = \arg\min_{\alpha_t^{\tilde{m}}} \sum_{t=1}^{T} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}})$.

Proof: *We first begin with*

$$(\alpha_{t+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2 = (\alpha_t^{\tilde{m}} - \eta \frac{\nabla_{\alpha_t^{\tilde{m}}} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}})}{\sqrt{\sum_{\tau=1}^{t} \|\nabla_{\alpha_\tau^{\tilde{m}}} L(\bar{\alpha}_\tau^{\tilde{m}}, \alpha_\tau^{\tilde{m}})\|_2^2}} - \alpha^{\tilde{m}})^2.$$

Then define adaptive learning rate η_t [10, 16] as

$$\eta_t := \frac{\eta}{\sqrt{\sum_{\tau=1}^{t} \|\nabla_{\alpha_\tau^{\tilde{m}}} L(\bar{\alpha}_\tau^{\tilde{m}}, \alpha_\tau^{\tilde{m}})\|_2^2}}.$$

So we have

$$(\alpha_{t+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2 = (\eta_t \nabla_{\alpha_t^{\tilde{m}}} L(\tilde{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}))^2 + (\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})^2 - 2\eta_t(\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})\nabla_{\alpha_t^{\tilde{m}}} L(\tilde{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}).$$

Therefore,

$$(\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})\nabla_{\alpha_t^{\tilde{m}}} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}) = \frac{(\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})^2 - (\alpha_{t+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2}{2\eta_t} + \frac{\eta_t}{2}(\nabla_{\alpha_t^{\tilde{m}}} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}))^2.$$

Since the loss function (3.1) *is convex, we have the following inequality*

$$L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}) - L(\bar{\alpha}_t^{\tilde{m}}, \alpha^{\tilde{m}}) \leq (\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})\nabla_{\alpha_t^{\tilde{m}}} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}). \tag{A.1}$$

By summing (A.1) *over $t \in [T]$ we have*

$$\begin{aligned}
&\sum_{t=1}^{T} \left(L\left(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}\right) - L\left(\bar{\alpha}_t^{\tilde{m}}, \alpha^{\tilde{m}}\right)\right) \\
&\leq \sum_{t=1}^{T} \frac{(\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})^2 - (\alpha_{t+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2}{2\eta_t} + \sum_{t=1}^{T} \frac{\eta_t}{2} \left(\nabla_{\alpha_t^{\tilde{m}}} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}})\right)^2 \\
&\leq \frac{\sqrt{T}}{2\eta} \sum_{t=1}^{T} \left((\alpha_t^{\tilde{m}} - \alpha^{\tilde{m}})^2 - (\alpha_{t+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2\right) + \frac{\eta}{2} \sum_{t=1}^{T} \frac{1}{\sqrt{\sum_{\tau=1}^{t} \|\nabla_{\alpha_\tau^{\tilde{m}}} L(\bar{\alpha}_\tau^{\tilde{m}}, \alpha_\tau^{\tilde{m}})\|_2^2}} \\
&\leq \frac{\sqrt{T}}{2\eta} \left((\alpha_1^{\tilde{m}} - \alpha^{\tilde{m}})^2 - (\alpha_{T+1}^{\tilde{m}} - \alpha^{\tilde{m}})^2\right) + \frac{\eta}{2} \sum_{t=1}^{T} \frac{1}{\alpha\sqrt{T}} \overset{(i)}{\leq} \frac{\sqrt{T}}{2\eta}(1 + 2\eta)^2 + \frac{\eta\sqrt{T}}{2\alpha},
\end{aligned} \tag{A.2}$$

where (i) used $\alpha_t^m \in [-\eta, 1 + \eta]$ by Lemma A.3.

Lemma A.2. *For miscoverage probability assigned to any model $\tilde{m} \in [M]$ we have the*

following bound

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\bar{w}_t^m L(\bar{\alpha}_t^m,\alpha_t^m)-\sum_{t=1}^{T}L(\bar{\alpha}_t^{\tilde{m}},\alpha_t^{\tilde{m}})\leq\frac{\ln M}{\epsilon}+\epsilon(1+\eta)^2T.$$

Proof: *Referring to the definition of* $\bar{w}_t^m$ *in Subsection 3.2.2 and defining* $W_t:=\sum_{m=1}^{M}w_t^m$, *we have*

$$\begin{aligned}
W_{T+1}&=\sum_{m=1}^{M}w_{T+1}^m=\sum_{m=1}^{M}w_T^m\exp\left(-\epsilon L(\bar{\alpha}_T^m,\alpha_T^m)\right)=W_T\sum_{m=1}^{M}\bar{w}_T^m\exp\left(-\epsilon L(\bar{\alpha}_T^m,\alpha_T^m)\right)\\
&\overset{(i)}{\leq} W_T\sum_{m=1}^{M}\bar{w}_T^m\left(1-\epsilon L(\bar{\alpha}_T^m,\alpha_T^m)+\epsilon^2L(\bar{\alpha}_T^m,\alpha_T^m)^2\right)\\
&=W_T\left(1-\epsilon\sum_{m=1}^{M}\bar{w}_T^mL(\bar{\alpha}_T^m,\alpha_T^m)+\epsilon^2\sum_{m=1}^{M}\bar{w}_T^mL(\bar{\alpha}_T^m,\alpha_T^m)^2\right)\\
&\overset{(ii)}{\leq} W_T\exp\left(-\epsilon\sum_{m=1}^{M}\bar{w}_T^mL(\bar{\alpha}_T^m,\alpha_T^m)+\epsilon^2\sum_{m=1}^{M}\bar{w}_T^mL(\bar{\alpha}_T^m,\alpha_T^m)^2\right)\\
&\leq W_1\exp\left(-\epsilon\sum_{t=1}^{T}\sum_{m=1}^{M}\bar{w}_t^mL(\bar{\alpha}_t^m,\alpha_t^m)+\epsilon^2\sum_{t=1}^{T}\sum_{m=1}^{M}\bar{w}_t^mL(\bar{\alpha}_t^m,\alpha_t^m)^2\right),
\end{aligned}\tag{A.3}$$

where $W_1=1$, (i) *follows from the inequality* $\exp(-\epsilon x)\leq 1-\epsilon x+\epsilon^2x^2$ *for* $|\epsilon|\leq 1$, *and* (ii) *follows from* $1+x\leq e^x$. *On the other hand, we have*

$$W_{T+1}\geq w_{T+1}^{\tilde{m}}=w_1^{\tilde{m}}\prod_{t=1}^{T}\exp\left(-\epsilon L(\bar{\alpha}_t^{\tilde{m}},\alpha_t^{\tilde{m}})\right)=w_1^{\tilde{m}}\exp\left(-\epsilon\sum_{t=1}^{T}L(\bar{\alpha}_t^{\tilde{m}},\alpha_t^{\tilde{m}})\right),\tag{A.4}$$

where $w_1^{\bar{m}}=\frac{1}{M}$. *By combining* (A.3) *and* (A.4) *we have*

$$\begin{aligned}
&\exp\left(-\epsilon\sum_{t=1}^{T}\sum_{m=1}^{M}\bar{w}_t^mL(\bar{\alpha}_t^m,\alpha_t^m)+\epsilon^2\sum_{t=1}^{T}\sum_{m=1}^{M}\bar{w}_t^mL(\bar{\alpha}_t^m,\alpha_t^m)^2\right)\\
&\geq\frac{1}{M}\exp\left(-\epsilon\sum_{t=1}^{T}L(\bar{\alpha}_t^{\tilde{m}},\alpha_t^{\tilde{m}})\right).
\end{aligned}\tag{A.5}$$

By taking the logarithm on both sides we have

$$-\epsilon \sum_{t=1}^{T} \sum_{m=1}^{M} \bar{w}_t^m L(\bar{\alpha}_t^m, \alpha_t^m) + \epsilon^2 \sum_{t=1}^{T} \sum_{m=1}^{M} \bar{w}_t^m L(\bar{\alpha}_t^m, \alpha_t^m)^2 \geq -\ln M - \epsilon \sum_{t=1}^{T} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}),$$

which leads to

$$\sum_{t=1}^{T} \sum_{m=1}^{M} \bar{w}_t^m L(\bar{\alpha}_t^m, \alpha_t^m) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{\tilde{m}}, \alpha_t^{\tilde{m}}) \leq \frac{\ln M}{\epsilon} + T\epsilon(1+\eta)^2. \tag{A.6}$$

Now, we define the best model in the static environment as

$$m^* = \arg\min_{m \in M} \sum_{t=1}^{T} L(\bar{\alpha}_t^m, \alpha_t^m).$$

Then, we replace $\tilde{m}$ with best model m^* in Lemma A.1 and A.2.By setting $\epsilon = \frac{1}{\sqrt{T}}$ and summing results of two lemmas we have:

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{m=1}^{M} \bar{w}_T^m L(\bar{\alpha}_t^m, \alpha_t^m) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha_t^{m^*}) + \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha_t^{m^*}) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha^{m^*}) \\
&= \sum_{t=1}^{T} \sum_{m=1}^{M} \bar{w}_T^m L(\bar{\alpha}_t^m, \alpha_t^m) - \sum_{t=1}^{T} L(\bar{\alpha}_t^{m^*}, \alpha^{m^*}) \leq \sqrt{T} \left(\frac{(1+2\eta)^2}{2\eta} + \frac{\eta}{2\alpha} + \ln M + (1+\eta)^2 \right).
\end{aligned} \tag{A.7}$$

Lemma A.3. *For every $\alpha \in [0,1]$ and learning rate $\eta > 0$, adaptive miscoverage probability α_t^m for any model $m \in [M]$ and $t \in [T]$ is bounded as*

$$\alpha_t^m \in [-\eta, 1+\eta].$$

Proof of lemma A.3 Based on equation (3.3), we have

$$|\alpha_t^m - \alpha_{t-1}^m| = \eta \left| \frac{(\alpha - err_{t-1}^m)}{\sqrt{\sum_{\tau=1}^{t-1} \|\nabla_{\alpha_\tau^m} L(\bar{\alpha}_\tau^m, \alpha_\tau^m)\|_2^2}} \right| \leq \eta. \tag{A.8}$$

We prove this lemma using a contradiction. Suppose there exists a $\hat{t}$ such that $\alpha_{\hat{t}}^m \notin [-\eta, 1+\eta]$, where $\hat{t} \geq 2$ is the smallest such time index. We first prove the case of violating the upper bound; by contradiction, assume $\alpha_{\hat{t}}^m > 1+\eta$. According to equation (A.8), this would necessitate that $\alpha_{\hat{t}-1}^m \geq 1$. Given that we assumed $\hat{t}$ is the smallest time index to violate the upper bound, it should follow that $\alpha_{\hat{t}-1}^m \leq 1+\eta$. However, $\alpha_{\hat{t}-1}^m > 1 > \bar{\alpha}_{\hat{t}-1}^m$ implies $err_{\hat{t}-1}^m = 1$. By equation (3.3) we have

$$\alpha_{\hat{t}}^m = \alpha_{\hat{t}-1}^m + \eta \frac{\alpha - 1}{\sqrt{\sum_{\tau=1}^{\hat{t}-1} \|\nabla_{\alpha_\tau^m} L(\bar{\alpha}_\tau^m, \alpha_\tau^m)\|_2^2}} \leq \alpha_{\hat{t}-1}^m \leq 1+\eta,$$

which contradicts our assumption that $\alpha_{\hat{t}-1}^m > 1+\eta$. Next, assume $\alpha_{\hat{t}}^m < -\eta$. By equation (A.8), we have $\alpha_{\hat{t}-1}^m < 0$. Given that $\hat{t}$ is the smallest index that violates the lower bound of the lemma, it must hold that $\alpha_{\hat{t}-1}^m \geq -\eta$. Considering $\alpha_{\hat{t}-1}^m < 0 < \bar{\alpha}_{\hat{t}-1}^m$, we deduce that $err_{\hat{t}-1}^m = 0$. Therefore, by equation (3.3), we have

$$\alpha_{\hat{t}}^m = \alpha_{\hat{t}-1}^m + \eta \frac{\alpha}{\sqrt{\sum_{\tau=1}^{\hat{t}-1} \|\nabla_{\alpha_\tau^m} L(\bar{\alpha}_\tau^m, \alpha_\tau^m)\|_2^2}} \geq \alpha_{\hat{t}-1}^m \geq -\eta,$$

which contradicts our initial assumption that $\alpha_{\hat{t}}^m < -\eta$.

A.2 Proof of Theorem 4.1

We first define the expected miscoverage error as

$$\mathbb{E}[err_t] = \sum_{n=1}^{t}\sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} err_t^{mn}.$$

The proof of this theorem is based on a grouping argument. So we first divide T into $\lceil T^{1-\gamma} \rceil$ group for $\gamma \in (\frac{1}{2}, 1)$, and write the kth group as

$$G_k = \{t_{k-1}+1, \ldots, \min(t_k, T)\}.$$

where $|G_k| \leq \lceil T^{\gamma} \rceil$. We also define a new variable, $H_{n:t}^{mn}$, assigned to mth update rule of nth expert as follows

$$H_{n:t}^{mn} := \sqrt{\sum_{\tau=n}^{t} \|\nabla_{\alpha_\tau^{mn}} L(\bar{\alpha}_\tau^{mn}, \alpha_\tau^{mn})\|_2^2}.$$

So the update rule in (3.3) can be written as:

$$\alpha_{t+1}^{mn} = \alpha_t^{mn} + \eta \frac{(\alpha - err_t^{mn})}{H_{n:t}^{mn}}. \tag{A.9}$$

For kth group where $2 \leq k \leq \lceil T^{1-\gamma} \rceil$, by using (A.9) we have

$$\mathbb{E}[err_t] - \alpha = \sum_{n=1}^{t}\sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} (err_t^{mn} - \alpha) = \frac{1}{\eta} \sum_{n=1}^{t}\sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} (\alpha_t^{mn} - \alpha_{t+1}^{mn}) H_{n:t}^{mn}.$$

Since at each time t we activate an expert with lifetime as defined in (4.1), the nth expert will be activated at time $t = n$. Consequently, $\bar{h}_t^n$ will be 0 for $t < n$. Therefore, we have

$$\begin{aligned}
&\frac{1}{\eta}\sum_{n=1}^{t}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t}^{mn} = \frac{1}{\eta}\sum_{n=1}^{t_{k-1}}\sum_{m=1}^{M}\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t_{k-1}}^{mn}\\
&+\frac{1}{\eta}\sum_{n=1}^{t}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t}^{mn} - \frac{1}{\eta}\sum_{n=1}^{t_{k-1}}\sum_{m=1}^{M}\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t_{k-1}}^{mn}\\
&=\frac{1}{\eta}\sum_{n=1}^{t_{k-1}}\sum_{m=1}^{M}\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t_{k-1}}^{mn}\\
&+\frac{1}{\eta}\sum_{n=1}^{t}\sum_{m=1}^{M}(\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}})(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t}^{mn}. \qquad \text{(A.10)}
\end{aligned}$$

Note that according to (A.9), $H_{n:t}^{mn}(\alpha_t^{mn}-\alpha_{t+1}^{mn}) \le \eta$. By summing (A.10) over $t \in G_k$ we have

$$\begin{aligned}
&\left|\sum_{t\in G_k}(\mathbb{E}[err_t]-\alpha)\right| \le \left|\frac{1}{\eta}\sum_{n=1}^{t_{k-1}}\sum_{m=1}^{M}\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}H_{n:t_{k-1}}^{mn}\sum_{t\in G_k}(\alpha_t^{mn}-\alpha_{t+1}^{mn})\right|\\
&+\left|\frac{1}{\eta}\sum_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}(\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}})(\alpha_t^{mn}-\alpha_{t+1}^{mn})H_{n:t}^{mn}\right|\\
&\le\left|\frac{1}{\eta}\sum_{n=1}^{t_{k-1}}\sum_{m=1}^{M}\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}H_{n:t_{k-1}}^{mn}(\alpha_{t_{k-1}+1}^{mn}-\alpha_{t_k+1}^{mn})\right|\\
&+\left|\sum_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}(\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}})\right|\\
&\le\frac{1}{\eta}\max_{\{m\in[M],n\in[t_{k-1}]\}}H_{n:t_{k-1}}^{mn}\left|\alpha_{t_{k-1}+1}^{mn}-\alpha_{t_k+1}^{mn}\right|\\
&+|G_k|\cdot\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right|\\
&\le\frac{1+2\eta}{\eta}\sqrt{T}+\lceil T^\gamma\rceil\cdot\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right| \qquad \text{(A.11)}
\end{aligned}$$

For G_1 we have

$$\left|\sum_{t\in G_1}\mathbb{E}[err_t]-\alpha\right| = \sum_{t\in G_1}\sum_{n=1}^{t}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}(err_t^{mn}-\alpha) \le \sum_{t\in G_1}\sum_{n=1}^{t}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{m_n} \le |G_1| \le \lceil T^\gamma\rceil. \tag{A.12}$$

By summing over all group we have

$$\begin{aligned}
&\left|\sum_{t=1}^{T}\mathbb{E}[err_t]-\alpha\right| = \sum_{k=1}^{\lceil T^{1-\gamma}\rceil}\left|\sum_{t\in G_k}\mathbb{E}[err_t]-\alpha\right| \\
&\le 2T^\gamma + \sum_{k=2}^{\lceil T^{1-\gamma}\rceil}\left(\frac{1+2\eta}{\eta}\sqrt{T}+2T^\gamma\cdot\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right|\right) \\
&\le T^{\frac{3}{2}-\gamma}\frac{1+2\eta}{\eta}+2T^\gamma\left(1+\sum_{k=2}^{\lceil T^{1-\gamma}\rceil}\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right|\right) \\
&\le \mathcal{O}(T^{\frac{3}{2}-\gamma}+T^\gamma\left(1+\sum_{k=2}^{\lceil T^{1-\gamma}\rceil}\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right|\right)
\end{aligned} \tag{A.13}$$

We define $\beta_\gamma(T)$ as:

$$\beta_\gamma(T) := \left(1+\sum_{k=2}^{\lceil T^{1-\gamma}\rceil}\max_{t\in G_k}\sum_{n=1}^{t}\sum_{m=1}^{M}\left|\bar{h}_t^n\bar{w}_t^{mn}-\bar{h}_{t_{k-1}}^n\bar{w}_{t_{k-1}}^{mn}\frac{H_{n:t_{k-1}}^{mn}}{H_{n:t}^{mn}}\right|\right). \tag{A.14}$$

Then we have:

$$\left|\sum_{t=1}^{T}\mathbb{E}[err_t]-\alpha\right| = \mathcal{O}(T^{\frac{3}{2}-\gamma}+T^\gamma\beta_\gamma(T)).$$

A.3 Proof of Theorem 4.2

We can write the regret as

$$\sum_{t\in I_{\tilde{n}}}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t\in I_{\tilde{n}}}\sum_{m=1}^{M}\bar{w}_t^{m\tilde{n}}L(\bar{\alpha}_t^{m\tilde{n}},\alpha_t^{m\tilde{n}})$$
$$+\sum_{t\in I_{\tilde{n}}}\sum_{m=1}^{M}\bar{w}_t^{m\tilde{n}}L(\bar{\alpha}_t^{m\tilde{n}},\alpha_t^{m\tilde{n}})-\sum_{t\in I_{\tilde{n}}}L(\bar{\alpha}_t^{m^*\tilde{n}},\alpha^{m^*\tilde{n}}), \tag{A.15}$$

where $I_{\tilde{n}}$ denotes the time interval during which expert $\tilde{n}$ is active, starting at time $t=\tilde{n}$. The third and fourth terms in the expression are analogous to the regret experienced by expert $\tilde{n}$, as established in Theorem 3.1. To evaluate the regret for the first and second terms, we employ Lemma A.2. The main difference is in the number of experts considered: For a looser bound, assuming that experts remain active beyond their designated lifetime, the maximum number of experts at each time step t would be gt. Thus, we derive the following bound for the first and second terms

$$\sum_{t\in I_{\tilde{n}}}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{m\tilde{n}}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t\in I_{\tilde{n}}}\sum_{m=1}^{M}\bar{w}_t^{m\tilde{n}}L(\bar{\alpha}_t^{m\tilde{n}},\alpha_t^{m\tilde{n}})$$
$$\leq\frac{\sqrt{|I_{\tilde{n}}|}\ln(gt)}{\sigma}+\frac{|I_{\tilde{n}}|\sigma(1+\eta)^2}{\sqrt{|I_{\tilde{n}}|}}\leq\sqrt{|I_{\tilde{n}}|}\left(\ln(gt)+\sigma^2(1+\eta)^2\right). \tag{A.16}$$

By summing (A.16) with regret bound of Theorem 3.1 we have

$$\sum_{t\in I_{\tilde{n}}}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t\in I_{\tilde{n}}}L(\bar{\alpha}_t^{m^*\tilde{n}},\alpha^{m^*\tilde{n}})$$
$$\leq\sqrt{|I_{\tilde{n}}|}\left(\frac{(1+2\eta)^2}{2\eta}+\frac{\eta}{2\alpha}+\ln M+(1+\eta)^2\right)+\sqrt{|I_{\tilde{n}}|}\left(\ln(gt)+\sigma^2(1+\eta)^2\right)$$
$$=\sqrt{|I_{\tilde{n}}|}\left(\frac{(1+2\eta)^2}{2\eta}+\frac{\eta}{2\alpha}+\ln M+\ln g+(1+\sigma^2)(1+\eta)^2+\ln t\right), \tag{A.17}$$

Until this point, the regret bound we've established applies solely to intervals that start with expert $\tilde{n}$, where each interval's length corresponds to the lifetime of expert $\tilde{n}$. However, to derive a regret bound for any arbitrary time interval I, we need to partition the interval into subintervals in a suitable manner. As proposed in [8], we can divide an interval I into two sequences of non-overlapping and consecutive intervals, denoted as $(I_{-p}, ..., I_0)$ and $(I_1, ..., I_q)$, such that $\frac{|I_{r+1}|}{|I_r|} \leq \frac{1}{2}$ for all $r \in (1, q-1)$, and $\frac{|I_r|}{|I_{r+1}|} \leq \frac{1}{2}$ for all $r \in (-p, -1)$. Subsequently, by employing the inequality $\sum_{r=1}^{\infty} \sqrt{2^{-r} T_0} \leq 4\sqrt{T_0}$, and by replacing $\tilde{n}$ with n^* using (4.5), we have

$$
\begin{aligned}
&\sum_{t\in I} \sum_{n\in\mathcal{A}(t)} \sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) - \sum_{t\in I} L(\bar{\alpha}_t^{m^*n^*}, \alpha^{m^*n^*}) \\
&= \sum_{r=1}^{q-1} \sum_{t\in I_r} \sum_{n\in\mathcal{A}(t)} \sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) - \sum_{r=1}^{q-1} \sum_{t\in I_r} L(\bar{\alpha}_t^{m^*n^*}, \alpha^{m^*n^*}) \\
&+ \sum_{r=-p}^{-1} \sum_{t\in I_r} \sum_{n\in\mathcal{A}(t)} \sum_{m=1}^{M} \bar{h}_t^n \bar{w}_t^{mn} L(\bar{\alpha}_t^{mn}, \alpha_t^{mn}) - \sum_{r=-p}^{-1} \sum_{t\in I_r} L(\bar{\alpha}_t^{m^*n^*}, \alpha^{m^*n^*}) \\
&\leq A\sqrt{|I|} + B \ln T \sqrt{|I|}. \qquad \text{(A.18)}
\end{aligned}
$$

Given that A and B are positive constant variables, we have determined the regret bound for our problem, demonstrating sublinear regret for Algorithm 2.

A.4 Proof of Lemma 4.1

To prove the regret in a dynamic environment, we adopt a method which was first proposed by [4]. So we can write the dynamic regret in our problem as

$$\sum_{t=1}^{T}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t=1}^{T}L(\bar{\alpha}_t^{m^*n^*},\alpha^{m^*n^*})$$
$$+\sum_{t=1}^{T}L(\bar{\alpha}_t^{m^*n^*},\alpha^{m^*n^*})-\sum_{t=1}^{T}L(\bar{\alpha}_t^{m^*n^*},\alpha_t^{m^*n^*}), \tag{A.19}$$

where $\alpha_t^{m^*n^*}$ is obtained by (4.8). The first two terms in (A.19) represent the static regret as defined in Theorem 4.2 and have been shown to be bounded. Consequently, to establish the overall regret bound, it is sufficient to find the upper bounds for the third and fourth terms in (A.19). We begin by dividing the total time interval T into sub-intervals I_r indexed by $r=1,...,[T/|I|]$ where $|I|$ is the length of each interval. so we can rewrite (A.19) as

$$\sum_{r=1}^{[T/|I|]}\sum_{t\in I_r}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{r=1}^{[T/|I|]}\sum_{t\in I_r}L(\bar{\alpha}_t^{m^*n^*},\alpha^{m^*n^*})$$
$$+\sum_{r=1}^{[T/|I|]}\sum_{t\in I_r}L(\bar{\alpha}_t^{m^*n^*},\alpha^{m^*n^*})-\sum_{r=1}^{[T/|I|]}\sum_{t\in I_r}L(\bar{\alpha}_t^{m^*n^*},\alpha_t^{m^*n^*}). \tag{A.20}$$

For the second two terms we have

$$\sum_{t\in I_r}L(\bar{\alpha}_t^{m^*n^*},\alpha^{m^*n^*})-\sum_{t\in I_r}L(\bar{\alpha}_t^{m^*n^*},\alpha_t^{m^*n^*})\leq 2|I|V(L(.)_{t\in I_r}),$$

where $V(L(.)_t)$ is the variability of environment [4] defined in (4.6). So the regret in (A.19) for any arbitrary $|I|$ will be

$$\sum_{t=1}^{T}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t=1}^{T}L(\bar{\alpha}_t^{m^*n^*},\alpha_t^{m^*n^*})$$
$$\leq\sum_{r=1}^{[T/|I|]}(A+B\ln T)\sqrt{|I|}+(2|I|V(L(.)_{t\in I_r})=(A+B\ln T)\frac{T}{\sqrt{|I|}}+(2|I|V(L(.)_{t=1}^{T})). \tag{A.21}$$

Since (A.18) holds for any interval $I\subseteq[T]$, By selecting $|I|=\left|(\frac{T}{V(L(.)_{t=1}^{T})})^{\frac{2}{3}}\right|$ we have

$$\sum_{t=1}^{T}\sum_{n\in\mathcal{A}(t)}\sum_{m=1}^{M}\bar{h}_t^n\bar{w}_t^{mn}L(\bar{\alpha}_t^{mn},\alpha_t^{mn})-\sum_{t=1}^{T}L(\bar{\alpha}_t^{m^*n^*},\alpha_t^{m^*n^*})$$
$$(A+B\ln T)T^{\frac{2}{3}}V^{\frac{1}{3}}(L(.)_{t=1}^{T})+2T^{\frac{2}{3}}V^{\frac{1}{3}}(L(.)_{t=1}^{T})\leq\tilde{\mathcal{O}}(T^{\frac{2}{3}}V^{\frac{1}{3}}(L(.)_{t=1}^{T})) \tag{A.22}$$

There are also analyses for bounding the deviation from time-varying optima from $t=1$ to $t=T$ in decentralized and federated optimization contexts, including settings with local updates and gradient tracking [12], and low-rank fine-tuning [11].