ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

LEARNING GENERAL TYPE-2 FUZZY LOGIC SYSTEMS FOR UNCERTAINTY QUANTIFICATION

M.Sc. THESIS

Yusuf GÜVEN

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JULY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

LEARNING GENERAL TYPE-2 FUZZY LOGIC SYSTEMS FOR UNCERTAINTY QUANTIFICATION

M.Sc. THESIS

Yusuf GÜVEN
(504231132)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor : Prof. Dr. Tufan KUMBASAR

JULY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BELİRSİZLİK NİCELLEŞTİRİLMESİ İÇİN GENEL TİP-2 BULANIK MANTIK SİSTEMLERİNİN ÖĞRENİLMESİ

YÜKSEK LİSANS TEZİ

Yusuf GÜVEN
(504231132)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı : Prof. Dr. Tufan KUMBASAR

TEMMUZ 2025

Yusuf GÜVEN, a M.Sc. student of ITU Graduate School student ID 504231132, successfully defended the thesis entitled “LEARNING GENERAL TYPE-2 FUZZY LOGIC SYSTEMS FOR UNCERTAINTY QUANTIFICATION”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Tufan KUMBASAR** ....
Istanbul Technical University

Jury Members : **Assoc. Dr. İlker ÜSTOĞLU** ....
Istanbul Technical University

Dr. Alec STOTHERT ....
MathWorks, Inc.

Date of Submission : 05 June 2025
Date of Defense : 16 July 2025

To my mom,

FOREWORD

I would like to express my deepest gratitude to my family, who have always provided me with an environment full of comfort and encouragement throughout the challenges I faced during my thesis journey. Their unwavering support lightened my burdens and gave me the courage to move forward. Without their presence, this thesis, which is dedicated to them, would not have been possible.

I am sincerely grateful to my thesis advisor, Prof. Dr. Tufan KUMBASAR, for his continuous guidance, insightful advice, and immense knowledge. His mentorship and support allowed me to work on advanced topics in uncertainty quantification and fuzzy sets and systems within his dedicated research group.

I would also like to express my heartfelt thanks to MathWorks for their generous support through the MathWorks Research Scholarship. I extend my sincere gratitude specifically to Dr. Alec Stothert and Dr. Rajibul Huq from MathWorks for their valuable guidance, insightful discussions, and support throughout the project. The collective contributions from MathWorks greatly facilitated the progress and depth of my research.

July 2025 Yusuf GÜVEN

TABLE OF CONTENTS

Page

ABBREVIATIONS

DL	**:** Deep Learning
FLS	**:** Fuzzy Logic System
FOU	**:** Footprint of Uncertainty
GT2	**:** General Type-2
IT2	**:** Interval Type-2
PI	**:** Prediction Interval
HQ-PI	**:** High Quality Prediction Interval
FS	**:** Fuzzy Set
MJ-GT2-FS	**:** Mendel and John’s General Type-2 Fuzzy Set
Z-GT2-FS	**:** Zadeh’s General Type-2 Fuzzy Set
MJ-GT2-FLS	**:** Mendel and John’s General Type-2 Fuzzy Logic System
Z-GT2-FLS	**:** Zadeh’s General Type-2 Fuzzy Logic System
PMF	**:** Primary Membership Function
SMF	**:** Secondary Membership Function
T1	**:** Type-1
T2	**:** Type-2
UQ	**:** Uncertainty Quantification
TRS	**:** Type-Reduced Set
α**-IT2-FLS**	**:** α-plane associated IT2-FLS
QR	**:** Quantile Regression
SQR	**:** Simultaneous Quantile Regression
ASQR	**:** Adaptive Simultaneous Quantile Regression
AD	**:** Automatic Differentiation
LP	**:** Learnable Parameter
LMF	**:** Lower Membership Function
IQR	**:** Inter Quartile Range
UMF	**:** Upper Membership Function

SYMBOLS

$\tilde{A}_{p,m}$	: General Type-2 Fuzzy Set.
$\tilde{A}_{p,m}^{\alpha_k}$	: α-plane of $\tilde{A}_{p,m}$ associated with α_k.
y	: The output of the General Type-2 Fuzzy Logic System.
y^{α_k}	: The output of the α_k- Interval Type-2 Fuzzy Logic System.
$\underline{y}^{\alpha_k}$	: The lower output of the α_k-Interval Type-2 Fuzzy Logic System.
$\overline{y}^{\alpha_k}$	: The upper output of the α_k-Interval Type-2 Fuzzy Logic System.
$\underline{f}_p^{\alpha_k}$	: The α_k-plane associated lower rule firing of the p^{th} rule.
$\overline{f}_p^{\alpha_k}$	: The α_k-plane associated upper rule firing of the p^{th} rule.
$\underline{\mu}_{\tilde{A}^{\alpha_k}}$	: The α_k-plane associated lower membership grade.
$\overline{\mu}_{\tilde{A}^{\alpha_k}}$	: The α_k-plane associated upper membership grade.
J_x	: Primary Membership Function of x.
c	: Center of Primary Membership Function.
$\underline{\sigma}$	: Lower Standard Deviation of Primary Membership Function.
$\overline{\sigma}$	: Upper Standard Deviation of Primary Membership Function.
$\delta_{0,1}$	: Parameters defining Secondary Membership Function.
γ	: Center of Secondary Membership Function.
σ^l	: Left Standard Deviation of Secondary Membership Function.
σ^r	: Right Standard Deviation of Secondary Membership Function.
$\underline{\tau}$	: Lower Quantile Level.
$\overline{\tau}$	: Upper Quantile Level.

LIST OF TABLES

LIST OF FIGURES

LEARNING GENERAL TYPE-2 FUZZY LOGIC SYSTEMS FOR UNCERTAINTY QUANTIFICATION

SUMMARY

Deep learning has been widely used in various domains such as computer vision, natural language processing, large language models, autonomous driving, and robotics because it provides us with the flexibility to design complex architectures and achieve high performance. Consequently, we no longer hesitate to apply these models in high-risk areas like medical treatment and finance. However, these ambitions will fall short if our models yield unreliable outcomes under diverse conditions. In this context, uncertainty estimation becomes crucial: it tells us when to trust our predictions and helps us handle anomalies, outliers, and out-of-distribution examples.

In recent studies, different deep learning models such as bayesian neural networks, deep ensembles, monte carlo dropout, gaussian processes, and quantile regression have been used for uncertainty estimation. For example, bayesian neural networks model the weights of a neural network as probability distributions, providing uncertainty by capturing posterior distributions over weights. However, this approach comes with a high computational cost and stability issues on large-scale datasets. On the other hand, quantile regression is easy to implement with a single model and simple loss functions, and it also scales well with large datasets.

Type-2 fuzzy logic systems can be great candidates for estimating uncertainty. It has been shown that type-2 fuzzy logic systems are capable of handling uncertainties through their inherent structural model, which provides a degree of freedom, referred to as the footprint of uncertainty, for modeling these uncertainties. In recent studies, interval type-2 fuzzy logic systems, which are simplified versions of general type-2 fuzzy logic systems, have been used for modeling uncertainty while simultaneously generating highly accurate predictions. To achieve this, the type-reduced set of interval type-2 fuzzy logic systems is employed to estimate uncertainty through a pinball loss. On the other hand, the output of interval type-2 fuzzy logic systems is used for point-wise estimation with an appropriate empirical loss definition, resulting in a composite loss function. Furthermore, general type-2 fuzzy logic systems are also utilized to generate reliable prediction intervals and estimate highly accurate predictions by exploiting the shape and size of the secondary membership functions. It has been shown that using the secondary membership functions for point-wise predictions offers an efficient way to handle both uncertainty and accuracy. In most studies, general type-2 fuzzy sets, based on Mendel and John's definition, are widely used, although Zadeh first defined the concept of general type-2 fuzzy sets. This is due to the α-plane representation of general type-2 fuzzy sets, which facilitates the parameterization of the secondary membership function and demonstrates the equivalence between a general type-2 fuzzy logic system and a set of α-plane associated interval type-2 fuzzy logic systems. However, we identify some drawbacks in this definition, particularly regarding the direct dependency of the secondary membership functions on the primary membership functions. To define the secondary membership functions, the primary membership functions must first be defined. We believe this dependency could potentially reduce the learning performance of general type-2 fuzzy logic systems and also affect the design flexibility of general type-2 fuzzy sets negatively.

In this master's thesis, we revisit the definition of general type-2 fuzzy sets as originally defined by Zadeh. We first present Zadeh's definition of general type-2 fuzzy sets. This structure offers the flexibility to design the secondary membership functions of general type-2 fuzzy sets without depending on the primary membership functions. In this context, we propose the mathematical foundations of both the secondary and primary membership functions, each of which is a type-1 fuzzy set. Afterwards, to define the output of Zadeh's general type-2 fuzzy logic systems, we integrate the α-plane representation into Zadeh's general type-2 fuzzy sets. Subsequently, we define the α-cuts of the secondary membership function and extract the equivalent lower and upper membership functions corresponding to the α-planes of Zadeh's general type-2 fuzzy set. These membership grades are then directly used to calculate the output of the general type-2 fuzzy logic system, which is formulated based on the α-plane approach. This approach enhances modeling flexibility and learning efficiency. Furthermore, we develop a method to address the curse of dimensionality problem that arises in fuzzy logic systems due to the rule firing strengths. This method adjusts the primary membership grades based on the input dimensions, effectively overcoming the challenges associated with high-dimensional datasets. Additionally, we propose parameterization tricks to ensure that the definitions of general type-2 fuzzy sets are not violated. These tricks allow us to formulate an unconstrained optimization problem, which can be efficiently handled using deep learning optimizers and automatic differentiation methods.

We propose a deep learning framework to learn dual-focused Zadeh's general type-2 fuzzy logic systems. In this context, we first assign distinct roles to the interval type-2 fuzzy logic systems associated with each α_k-plane within a composite loss function. This loss function consists of two components, simultaneously focusing on uncertainty and accuracy. To address both aspects, we present two loss definitions, leveraging the shape and size of the secondary membership function. For both loss definitions, we use only the type-reduced set of the α_0-interval type-2 fuzzy logic system to learn the prediction interval by estimating the upper and lower quantile levels for a given confidence level in the uncertainty component of the composite loss function. On the other hand, for the accuracy component, we define two loss functions. For the first, we utilize the output of the general type-2 fuzzy logic system, and for the second, we use the output of the α_k-plane interval type-2 fuzzy logic system as a point-wise estimator. Then, we present the comparative performance analysis of Zadeh's general type-2 fuzzy logic systems on high-dimensional datasets by comparing them to their Mendel and John's general type-2 fuzzy logic systems and interval type-2 fuzzy logic systems counterparts. The statistical results show that Zadeh's general type-2 fuzzy logic systems can serve as an effective approach for achieving highly accurate point-wise estimations and generating high-quality prediction intervals, meaning narrow bands that capture uncertainty at a given coverage level.

We also present a deep learning framework based on Zadeh's general type-2 fuzzy logic systems to learn the inverse cumulative distribution function by estimating all quantile levels. This approach helps prevent the need for multiple training sessions for different desired coverage levels with given quantile pairs. Instead, any quantile pair can be selected to generate a prediction interval that provides the desired confidence level after one training section. In this context, we reformulate the output of the general type-2 fuzzy logic system by enforcing it to learn a specific quantile level, τ, through the assignment $\alpha = \tau$. In this way, each output of the α-plane associated interval type-2 fuzzy logic system is set to learn a quantile level function. To learn the inverse cumulative distribution with a general type-2 fuzzy logic system, we reformulate the simultaneous quantile regression by sampling random quantile levels. To enhance learning, we develop an approach called adaptive simultaneous quantile regression, which incorporates a miscalibration measure during training. This approach allows us to generate additional quantile levels from miscalibration areas, ensuring they are trained effectively using the general type-2 fuzzy logic system. Afterwards, we compare our method with state-of-the-art deep learning methods to show the superiority of our method.

BELİRSİZLİK NİCELLEŞTİRİLMESİ İÇİN GENEL TİP-2 BULANIK MANTIK SİSTEMLERİNİN ÖĞRENİLMESİ

ÖZET

Derin öğrenme, karmaşık mimariler tasarlama ve bu mimariler aracılığıyla yüksek performans seviyelerine ulaşma konusunda sunduğu olağanüstü esneklik sayesinde, günümüzde bilgisayarlı görü, doğal dil işleme, büyük dil modellerinin geliştirilmesi, otonom sürüş sistemleri ve ileri robotik uygulamaları gibi son derece çeşitli ve etki alanı geniş disiplinlerde yaygın bir şekilde kendine yer bulmuştur. Bu teknolojinin sağladığı ilerlemeler, daha önce çözülmesi güç olarak kabul edilen birçok probleme yenilikçi çözümler getirmiş ve bu alanlarda adeta bir paradigma değişimine yol açmıştır. Bu başarının doğal bir sonucu olarak, derin öğrenme modellerini, karar verme süreçlerinin kritik olduğu ve hataların ciddi sonuçlar doğurabileceği tıbbi tedavi protokollerinin belirlenmesi veya finansal piyasalardaki risk analizleri gibi yüksek risk taşıyan hassas alanlarda dahi uygulamaktan giderek daha az çekinir hale geldik. Ancak, bu iddialı ve umut verici hedeflere ulaşma çabalarımız, geliştirdiğimiz modellerin karşılaştıkları farklı ve beklenmedik koşullar altında tutarlı ve güvenilir sonuçlar üretememesi durumunda ne yazık ki başarısızlıkla sonuçlanma riski taşımaktadır. Tam da bu kritik noktada, belirsizlik tahmini kavramı hayati bir öneme sahip olmaktadır. Zira belirsizlik tahmini, modellerimizin ürettiği tahminlere ne ölçüde ve hangi koşullar altında güvenebileceğimiz konusunda bize değerli bilgiler sunmakla kalmaz, aynı zamanda sistemin normal çalışma koşullarının dışına çıkan anomalileri, veri kümesindeki genel dağılıma uymayan aykırı değerleri ve modelin daha önce karşılaşmadığı, eğitim veri setinin dağılımından farklı olan dağılım dışı örnekleri etkin bir şekilde tanımlamamıza ve yönetmemize olanak tanır. Bu sayede, modellerin güvenilirliği artırılır ve potansiyel riskler en aza indirilir.

Son yıllarda gerçekleştirilen akademik çalışmalarda, belirsizlik tahmininin önemi giderek daha fazla anlaşılmış ve bu alanda çeşitli derin öğrenme modelleri başarıyla kullanılmıştır. Bu modeller arasında Bayesci sinir ağları, derin topluluklar, Monte Carlo seyreltme tekniği, Gauss süreçleri ve kantil regresyonu gibi farklı yaklaşımlar öne çıkmaktadır. Örneğin, Bayesci sinir ağları, geleneksel sinir ağlarındaki deterministik ağırlıkların aksine, ağın ağırlıklarını olasılık dağılımları olarak modelleyerek bir belirsizlik ölçüsü sunar. Bu yaklaşım, ağırlıklar üzerindeki sonsal dağılımları yakalayarak modelin tahminlerindeki güven aralığını belirlemesine olanak tanır. Ancak, Bayesci sinir ağlarının bu sofistike yapısı, özellikle büyük ölçekli ve yüksek boyutlu veri kümeleriyle çalışıldığında, önemli bir hesaplama maliyeti ve eğitim sürecinde kararlılık sorunları gibi pratik zorlukları da beraberinde getirmektedir. Diğer bir popüler yaklaşım olan kantil regresyonu ise, tek bir model kullanılarak ve görece basit kayıp fonksiyonları tanımlanarak kolayca uygulanabilir olmasıyla dikkat çeker. Ayrıca, kantil regresyonu büyük veri kümeleriyle çalışırken ölçeklenebilirlik açısından da avantajlıdır ve farklı kantil değerleri için tahminler üreterek belirsizlik aralıkları oluşturulmasına imkan tanır.

Bu çeşitli yöntemlerin yanı sıra, tip-2 bulanık mantık sistemleri, belirsizlik tahmini konusunda dikkate değer ve güçlü adaylar olarak ön plana çıkmaktadır. Yapılan araştırmalar, tip-2 bulanık mantık sistemlerinin, "belirsizlik ayak izi" olarak adlandırılan ve modelin doğasında bulunan yapısal bir serbestlik derecesi sayesinde, karşılaşılan çeşitli belirsizlik türlerini etkili bir şekilde ele alma ve modelleme kapasitesine sahip olduğunu açıkça ortaya koymuştur. Özellikle son dönemdeki çalışmalarda, genel tip-2 bulanık mantık sistemlerinin daha basitleştirilmiş ve hesaplama açısından daha verimli versiyonları olan aralık tip-2 bulanık mantık sistemleri, bir yandan yüksek doğrulukta noktasal tahminler üretirken, diğer yandan da modelin tahminlerindeki belirsizliği başarılı bir şekilde modellemek amacıyla sıklıkla kullanılmıştır. Bu ikili amaca ulaşmak için, aralık tip-2 bulanık mantık sistemlerinin tip-indirgenmiş kümesi, genellikle bir pinball kayıp fonksiyonu aracılığıyla belirsizlik aralıklarını öğrenmek ve tahmin etmek için kullanılır. Eş zamanlı olarak, aralık tip-2 bulanık mantık sistemlerinin ürettiği net çıktı değeri, uygun bir ampirik kayıp fonksiyonu tanımıyla birleştirilerek noktasal tahminlerin doğruluğunu artırmak için kullanılır. Bu iki farklı amaç için tanımlanan kayıp fonksiyonlarının bir araya getirilmesiyle de bileşik bir kayıp fonksiyonu oluşturulur. Daha da ötesi, genel tip-2 bulanık mantık sistemleri, ikincil üyelik fonksiyonlarının sahip olduğu esnek şekil ve boyut özelliklerinden faydalanarak hem güvenilir tahmin aralıkları üretme hem de son derece doğru noktasal tahminler elde etme potansiyelini taşımaktadır. İkincil üyelik fonksiyonlarının noktasal tahminler için kullanılmasının, sistemin hem belirsizliği etkin bir şekilde yönetmesine hem de yüksek doğruluk seviyelerine ulaşmasına olanak tanıyan verimli bir strateji olduğu çeşitli çalışmalarla kanıtlanmıştır. Literatürdeki çoğu çalışmada, genel tip-2 bulanık kümeler kavramını ilk olarak Lotfi A. Zadeh tanımlamış olmasına rağmen, Mendel ve John tarafından önerilen tanıma dayanan genel tip-2 bulanık kümelerin daha yaygın olarak kullanıldığı görülmektedir. Bu durumun temel nedeni, genel tip-2 bulanık kümelerin α-düzlem gösteriminin sunduğu kolaylıklardır. Bu gösterim, ikincil üyelik fonksiyonunun parametrelendirilmesini önemli ölçüde basitleştirmekte ve bir genel tip-2 bulanık mantık sisteminin, bir dizi α-düzlem ilişkili aralık tip-2 bulanık mantık sistemine denk olduğunu göstermektedir. Ancak, Mendel ve John'un tanımında, özellikle ikincil üyelik fonksiyonlarının birincil üyelik fonksiyonlarına doğrudan ve kaçınılmaz bir şekilde bağımlı olması gibi bazı önemli dezavantajlar tespit ettik. Bu tanıma göre, ikincil üyelik fonksiyonlarını tanımlayabilmek için öncelikle birincil üyelik fonksiyonlarının belirlenmiş olması gerekmektedir. Bu sıkı bağımlılığın, genel tip-2 bulanık mantık sistemlerinin öğrenme performansını potansiyel olarak kısıtlayabileceğine ve aynı zamanda genel tip-2 bulanık kümelerin tasarım esnekliğini olumsuz yönde etkileyebileceğine inanmaktayız.

Bu yüksek lisans tez çalışmasında, genel tip-2 bulanık kümelerin tanımını, kavramın öncüsü olan Zadeh tarafından orijinal olarak ortaya konduğu şekliyle yeniden ele alıyor ve bu tanımın potansiyel avantajlarını araştırmayı hedefliyoruz. Bu doğrultuda, ilk olarak Zadeh'in genel tip-2 bulanık kümeler için önerdiği orijinal tanımı detaylı bir şekilde sunuyoruz. Bu tanımın getirdiği en önemli avantajlardan biri, genel tip-2 bulanık kümelerin ikincil üyelik fonksiyonlarının, birincil üyelik fonksiyonlarına herhangi bir zorunlu bağımlılık olmaksızın, daha serbest ve esnek bir şekilde tasarlanabilmesine olanak tanımasıdır. Bu bağlamda, her biri kendi başına birer tip-1 bulanık küme olan hem ikincil üyelik fonksiyonları hem de birincil üyelik fonksiyonları için gerekli matematiksel temelleri ve formülasyonları öneriyoruz. Ardından, Zadeh'in tanımına dayanan genel tip-2 bulanık mantık sistemlerinin çıktısını anlamlı bir şekilde tanımlayabilmek amacıyla, literatürde yaygın olarak kabul gören α-düzlem gösterimini Zadeh'in genel tip-2 bulanık kümelerine entegre ediyoruz. Bu entegrasyonun devamında, ikincil üyelik fonksiyonunun α-kesitlerini tanımlıyor ve bu kesitlerden hareketle, Zadeh'in genel tip-2 bulanık kümesinin farklı α-düzlemlerine karşılık gelen eşdeğer alt ve üst üyelik fonksiyonlarını matematiksel olarak türetiyoruz. Elde edilen bu üyelik dereceleri, α-düzlem yaklaşımına dayanılarak formüle edilen genel tip-2 bulanık mantık sisteminin nihai çıktısını hesaplamak için doğrudan ve etkin bir şekilde kullanılmaktadır. Bu yaklaşımın, modelleme esnekliğini artırdığına ve öğrenme verimliliğini olumlu yönde etkilediğine inanıyoruz. Ayrıca, bulanık mantık sistemlerinde, özellikle kural tabanının büyümesiyle birlikte ortaya çıkan ve "boyutsallık laneti" olarak bilinen önemli bir problemi ele almak üzere özgün bir yöntem geliştiriyoruz. Önerdiğimiz bu yöntem, girdi uzayının boyutlarına bağlı olarak birincil üyelik derecelerini dinamik bir şekilde ayarlamakta ve böylece yüksek boyutlu veri kümeleriyle çalışırken karşılaşılan zorlukların etkili bir şekilde üstesinden gelinmesine yardımcı olmaktadır. Ek olarak, genel tip-2 bulanık kümelerin matematiksel tanımlarının ve varsayımlarının ihlal edilmemesini garanti altına almak amacıyla çeşitli parametrelendirme hileleri öneriyoruz. Bu hileler, karmaşık kısıtlamalara sahip olabilecek optimizasyon problemini, kısıtsız bir optimizasyon problemine dönüştürmemize olanak tanımakta ve bu sayede derin öğrenme alanında yaygın olarak kullanılan optimize edicilerin ve otomatik farklılaştırma yöntemlerinin verimli bir şekilde uygulanabilmesini mümkün kılmaktadır.

Bu tez kapsamında, Zadeh'in tanımına dayanan ve aynı anda hem doğruluk hem de belirsizlik üzerine odaklanan, yani "çift odaklı" genel tip-2 bulanık mantık sistemlerini öğrenebilmek için kapsamlı bir derin öğrenme çerçevesi öneriyoruz. Bu çerçeve içerisinde, öncelikle bileşik bir kayıp fonksiyonu tanımlıyor ve bu fonksiyon dahilinde, her bir α_k-düzlemi ile ilişkilendirilmiş olan aralık tip-2 bulanık mantık sistemlerine farklı ve belirgin roller atıyoruz. Tanımladığımız bu bileşik kayıp fonksiyonu, temel olarak iki ana bileşenden oluşmaktadır: bunlardan ilki modelin tahminlerindeki belirsizliği, ikincisi ise tahminlerin doğruluğunu hedeflemektedir. Her iki önemli yönü de etkin bir şekilde ele alabilmek amacıyla, ikincil üyelik fonksiyonunun esnek şekil ve boyut özelliklerinden yararlanarak iki farklı kayıp tanımı sunuyoruz. Önerdiğimiz her iki kayıp tanımı için de, bileşik kayıp fonksiyonunun belirsizlik bileşeninde, belirli bir güven seviyesi (α_0) için üst ve alt kantil seviyelerini tahmin ederek güvenilir bir tahmin aralığı öğrenmek amacıyla, yalnızca α_0-düzlemiyle ilişkili aralık tip-2 bulanık mantık sisteminin tip-indirgenmiş kümesini kullanıyoruz. Diğer yandan, bileşik kayıp fonksiyonunun doğruluk bileşeni için ise iki alternatif kayıp fonksiyonu tanımlıyoruz. Bu alternatiflerden ilkinde, genel tip-2 bulanık mantık sisteminin doğrudan çıktısını bir noktasal tahminleyici olarak kullanırken; ikincisinde ise, belirli bir α_k-düzlemiyle ilişkili aralık tip-2 bulanık mantık sisteminin çıktısını noktasal tahminler için temel alıyoruz. Bu teorik altyapıyı oluşturduktan sonra, Zadeh'in tanımına dayanan genel tip-2 bulanık mantık sistemlerinin, özellikle yüksek boyutlu ve karmaşık veri kümeleri üzerindeki karşılaştırmalı performans analizini sunuyoruz. Bu analizde, önerdiğimiz sistemleri, literatürde yaygın olarak kullanılan Mendel ve John'un tanımına dayanan genel tip-2 bulanık mantık sistemleri ve daha basit yapıdaki aralık tip-2 bulanık mantık sistemleri gibi benzerleriyle kıyaslıyoruz. Elde edilen istatistiksel sonuçlar, Zadeh'in tanımına dayanan genel tip-2 bulanık mantık sistemlerinin, hem son derece yüksek doğrulukta noktasal tahminler elde etme hem de belirli bir kapsama düzeyinde belirsizliği etkin bir şekilde yakalayan dar ve dolayısıyla yüksek kaliteli tahmin aralıkları üretme konusunda etkili ve güçlü bir yaklaşım olarak hizmet edebileceğini açıkça göstermektedir.

Bu tez çalışmasında ayrıca, Zadeh'in genel tip-2 bulanık mantık sistemlerine dayanan ve veri dağılımının tüm kantil seviyelerini aynı anda tahmin ederek ters birikimli dağılım fonksiyonunu öğrenmeyi amaçlayan yenilikçi bir derin öğrenme çerçevesi daha sunuyoruz. Önerdiğimiz bu yaklaşımın en önemli avantajlarından biri, araştırmacıların veya uygulayıcıların farklı güven aralıkları veya kapsama düzeyleri için, her biri belirli bir kantil çiftine karşılık gelen çok sayıda ayrı eğitim süreci yürütme ihtiyacını ortadan kaldırmasıdır. Bunun yerine, model tek bir kapsamlı eğitim sürecinden geçtikten sonra, elde edilen ters birikimli dağılım fonksiyonu üzerinden istenilen herhangi bir kantil çifti seçilerek, arzu edilen güven düzeyini sağlayan bir tahmin aralığı kolaylıkla ve hızla oluşturulabilir. Bu esnekliği sağlamak amacıyla, genel tip-2 bulanık mantık sisteminin çıktı formülasyonunu, $\alpha = \tau$ şeklinde bir atama yaparak, sistemin belirli bir kantil seviyesi olan τ'yu öğrenmeye zorlayacak şekilde yeniden düzenliyoruz. Bu sayede, α-düzlemi ile ilişkilendirilmiş her bir aralık tip-2 bulanık mantık sisteminin çıktısı, farklı bir kantil seviyesine karşılık gelen bir fonksiyonu öğrenmek üzere ayarlanmış olur. Genel bir tip-2 bulanık mantık sistemi kullanarak ters birikimli dağılım fonksiyonunu etkin bir şekilde öğrenebilmek için, eğitim verilerinden rastgele kantil seviyeleri örnekleyerek eş zamanlı kantil regresyonu yaklaşımını yeniden formüle ediyor ve modelimize uyarlıyoruz. Öğrenme sürecini daha da iyileştirmek ve modelin özellikle zorlandığı bölgelerde daha iyi performans göstermesini sağlamak amacıyla, "uyarlanabilir eş zamanlı kantil regresyonu" adını verdiğimiz özgün bir yaklaşım geliştiriyoruz. Bu yaklaşım, eğitim süreci sırasında bir yanlış kalibrasyon ölçüsünü dinamik olarak dahil etmekte ve bu ölçüye dayanarak, modelin tahminlerinin gerçek değerlerden saptığı, yani yanlış kalibrasyonun yüksek olduğu bölgelerden ek kantil seviyeleri üretmemize olanak tanımaktadır. Bu sayede, üretilen bu ek kantil seviyelerinin genel tip-2 bulanık mantık sistemi tarafından daha etkili bir şekilde öğrenilmesi ve modelin genel performansının artırılması hedeflenmektedir. Son olarak, önerdiğimiz bu kapsamlı yöntemin üstünlüğünü ve etkinliğini kanıtlamak amacıyla, elde ettiğimiz sonuçları, literatürdeki en güncel ve en başarılı derin öğrenme tabanlı belirsizlik ve kantil tahmin yöntemleriyle kapsamlı bir şekilde karşılaştırıyor ve önerimizin avantajlarını ortaya koyuyoruz.

1. INTRODUCTION

Deep Learning (DL) has achieved remarkable breakthroughs across a wide range of applications, including computer vision, natural language processing, and robotics [1–6]. Despite these successes, a predominant focus on accuracy often overlooks a critical aspect of modern predictive modeling: Uncertainty Quantification (UQ) [7–10]. In settings where decisions carry significant consequences, understanding and quantifying the uncertainty in predictions is as crucial as achieving high accurate results [11–13].

Fuzzy Logic Systems (FLSs) provide a powerful and flexible framework for handling uncertainty, making them well-suited for applications where uncertainty is significant. In the context of Uncertainty Quantification (UQ), Type-2 (T2) FLSs are particularly advantageous because they incorporate uncertainty directly into their Membership Functions (MFs). These MFs can be represented using either General Type-2 (GT2) Fuzzy Sets (FSs), which provide the most comprehensive modeling of uncertainty, or their simplified variant, Interval Type-2 (IT2) FSs [14]. GT2-FSs were originally introduced by Zadeh (Z) [15, 16] and have since been widely studied and utilized, particularly through the formalization provided by Mendel and John (MJ) [17]. The MJ-GT2-FSs framework is implemented via the zSlices/α-plane representation, which offers a structured approach to handling the complexities of GT2-FSs [14, 18]. This representation is built upon two fundamental principles:

1. *Parameterized Secondary MFs (SMFs):* The Secondary MF (SMF) of a GT2-FS is parameterized based on its Primary MF (PMF), allowing for a more structured and interpretable representation of uncertainty.

2. *Equivalence with IT2-FLSs:* The α-plane representation establishes a direct connection between GT2-FLSs and a set of associated IT2-FLSs, effectively enabling the decomposition of complex GT2 FLSs into a set of IT2-FLSs.

The learning procedures of IT2 and GT2 FLSs have traditionally focused on improving accuracy across various applications [1, 19–26]. Most studies have concentrated on refining MF structures, optimizing parameter learning algorithms, and enhancing rule-based inference techniques to achieve better predictive performance [1, 19–26]. While these advancements have led to significant improvements in accuracy, they have largely overlooked the potential of IT2 and GT2-FSs for UQ. Given the inherent ability of IT2 and GT2-FSs to model uncertainty within their structure, their application in UQ remains an underexplored yet promising area of research.

Recently, there has been a growing interest in using T2 FSs for UQ[27–30]. In [29], a DL framework is introduced using a composite loss function commonly applied in Quantile Regression (QR). Their approach explicitly incorporates the Type-Reduced Set (TRS) and the output of IT2-FLSs to train models that can generate Prediction Intervals (PIs). Similarly, for MJ-GT2-FLS [30], a different composite loss function is designed to utilize the support of SMFs (i.e., PMFs) for learning PIs while shaping the model for point-wise estimation. Both methods aim to learn PIs by ensuring a specific quantile range is covered, rather than capturing the entire predictive distribution—an objective that is considered the most comprehensive approach to UQ [12, 31].

In this thesis, we explore GT2 FSs based on Zadeh's definition to achieve high performance in both accuracy and uncertainty modeling through GT2-FLSs. Specifically, we adopt Z-GT2-FS instead of MJ-GT2-FS to eliminate the dependency of the secondary membership grade on the primary membership grade. This provides greater design flexibility, making the system more adaptable. We begin by establishing the mathematical foundations of SMF and PMF, both of which are defined using Type-1 (T1) FSs. Next, we incorporate the α-plane representation into Z-GT2-FSs to formally define the output of Z-GT2-FLSs. This involves defining the α-cut representation of SMF and extracting the Lower MF (LMF) and Upper MF (UMF) for each corresponding α-plane of Z-GT2-FSs. These extracted MFs directly contribute to computing the final output of Z-GT2-FLSs using the α-plane representation. To address the curse of dimensionality during the learning process of Z-GT2-FLSs, we introduce an approach that dynamically adjusts the PMF based on the input dimension. Additionally, we propose parameterization tricks that ensure the integrity of Z-GT2-FS definitions. This allows us to train Z-GT2-FLSs effectively using standard unconstrained DL optimizers and Automatic Differentiation (AD), making the learning process more efficient and scalable.

After formulating Z-GT2-FLSs as a structure trainable via DL optimizers, we introduce a DL framework that employs distinct α_k-planes (α_k-IT2-FLSs) within a composite loss function. This framework is designed to generate both high-quality Prediction Intervals (HQ-PIs) [9] and accurate point-wise predictions simultaneously. To achieve this, we propose two distinct loss functions that leverage the shape and size of the SMF in Z-GT2-FSs. The composite loss function consists of two main components: an uncertainty term and an accuracy term. The uncertainty term utilizes the TR set of the α_0-plane (α_0-IT2-FLSs) to generate PIs by estimating lower and upper quantile levels $(\underline{\tau}, \overline{\tau})$ for a given confidence level. Meanwhile, the accuracy term introduces two approaches: (1) using the aggregated output of Z-GT2-FLSs and (2) employing the output of the α_K-plane (α_K-IT2-FLSs) for point-wise estimation. To evaluate the effectiveness of Z-GT2-FLSs, we conduct extensive experiments on high-dimensional datasets, comparing their learning performance with MJ-GT2-FLSs and IT2-FLSs [29, 30], both of which involve a higher number of learnable parameters (LPs). Statistical analyses demonstrate that Z-GT2-FLSs provide a promising solution for achieving both high prediction accuracy and reliable HQ-PIs for a given confidence level.

Additionally, we propose a DL framework based on Z-GT2-FLSs to learn the inverse cumulative distribution function by estimating all quantile levels simultaneously. This approach enables us to capture the entire conditional distribution of the target variable, allowing for the selection of appropriate quantile levels for any given confidence level after one training section. To achieve this, we reformulate the GT2-FLS output as $y(\boldsymbol{x}, \alpha_k)$, ensuring that it learns a specific quantile level τ by setting $\alpha = \tau$. As a result, each α-IT2-FLS is designed to approximate a quantile function, offering the flexibility to generate any desired $\tau \in [0, 1]$ by adjusting α within the same range. To model the predictive distribution using GT2-FLS, we adapt the Simultaneous Quantile Regression (SQR) approach [12], which involves sampling random quantile levels. To further enhance the learning process, we introduce an Adaptive SQR (ASQR) method that incorporates a miscalibration measure for improved accuracy. Finally, we evaluate the estimation performance of GT2-FLS in comparison with DL-based methods to demonstrate its advantages.

The organization of this thesis consist of 6 chapters. In Chapter 2, we review some mathematical foundations on that are utilized throughout the thesis. The topics we review include fundamental definitions from the theory of IT2/GT2-FLSs. In Chapter

3, we motivate and present the Z-GT2-FLSs and explain its key differences from MJ-GT2-FLSs; we additionally propose the solutions Z-GT2-FLSs produce to the problems of IT2-FLSs. Chapter 4 describes the learning method for dual-focused Z-GT2-FLSs and shows the statistical analysis of Z-GT2-FLSs compared to MJ-GT2, and IT2-counterparts. In Chapter 5, we explore how Z-GT2-FLSs can be leveraged for predictive distribution estimation by assigning α-planes to quantile levels (τ). In Chapter 6, we conclude with an overview of Z-GT2-FLSs' applications and future work.

2. BACKGROUND: MATHEMATICAL PRELIMINARIES

In this chapter, we will first review the mathematical fundamentals of Type-2 Fuzzy Sets, providing a foundation for understanding their theoretical framework. After establishing these fundamental concepts, we will then explain the structure and working principles of IT2 and GT2-FLSs.

2.1 Type-2 Fuzzy Sets

A Type-2 Fuzzy Set can be defined as follows [32]:

$$\tilde{A} = \left\{(x,u), \mu_{\tilde{A}}(x,u) \mid x \in X, u \in U\right\} \tag{2.1}$$

Here, x represents the input variable, also referred to as the primary variable of $\tilde{A}$, while X denotes its corresponding universe[32]. The variable u serves as the secondary variable of $\tilde{A}$, with $U \equiv [0,1]$ representing its universe. Lastly, $\mu_{\tilde{A}}(x,u)$ defines the T2-MF of the T2-FS $\tilde{A}$. $\tilde{A}$ can also be expressed in the continuous and the discrete universe as follows:

$$\tilde{A} = \int_{x \in X} \int_{u \in U} \mu_{\tilde{A}}(x,u) \,/\, (x,u) \tag{2.2}$$

$$\tilde{A} = \sum_{x \in X} \sum_{u \in U} \mu_{\tilde{A}}(x,u) \,/\, (x,u) \tag{2.3}$$

After defining a T2-FS denoted by $\tilde{A}$, we can define the 2-D support of $\mu_{\tilde{A}}(x,u)$ (which is called Footprint of Uncertainty (FOU) of $\tilde{A}$) as follows [32]:

$$\text{FOU}(\tilde{A}) = \{(x,u) \in X \times [0,1],\ \mu_{\tilde{A}}(x,u) > 0\} \tag{2.4}$$

FOU is bounded by Lower MF (LMF) and Upper MF (UMF), which are denoted by $\underline{\mu}_{\tilde{A}}(x,u)$ and $\overline{\mu}_{\tilde{A}}(x,u)$ respectively and described as follows [32]:

$$\text{LMF}(\tilde{A}) = \underline{\mu}_{\tilde{A}}(x,u) = \inf\left\{u \mid u \in [0,1], \mu_{\tilde{A}}(x,u) > 0\right\} \tag{2.5}$$

$$\text{UMF}(\tilde{A}) = \overline{\mu}_{\tilde{A}}(x,u) = \sup\left\{u \mid u \in [0,1], \mu_{\tilde{A}}(x,u) > 0\right\} \tag{2.6}$$

The primary membership at $x \in X$ of the given T2-FS ($\tilde{A}$) is described as J_x and defined by the interval of $[\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)]$ as[32]:

$$J_x = \left\{u \in [0,1], \mu_{\tilde{A}}(x,u) > 0\right\} = [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)] \tag{2.7}$$

The secondary membership at $x \in X$ of the given T2-FS ($\tilde{A}$) is described as $\mu_{\tilde{A}(x)}(u)$ or $\mu_{\tilde{A}(x)}$ and defined as follows[32]:

$$\mu_{\tilde{A}(x)}(u) = \mu_{\tilde{A}(x)} = \mu_{\tilde{A}_x} = \int\limits_{u \in [0,1]} \mu_{\tilde{A}}(x,u) \; /u \tag{2.8}$$

2.1.1 Interval type-2 fuzzy sets

An Interval Type-2 fuzzy set is defined as

$$\tilde{A} = \left\{(x,u), \, \mu_{\tilde{A}}(x,u) = 1 \mid x \in X, \; u \in U\right\}, \tag{2.9}$$

where the secondary membership function $\mu_{\tilde{A}}(x,u)$ maps each primary element $x \in X$ and secondary variable $u \in U$ to 1. An IT2-FS is a special case of a GT2-FS all of whose secondary grades are equal to 1[32]. An IT2-MF is deployed in the 2D domain instead of 3D, differing from GT2-MF. In this context, an IT2-FS is fully defined by its LMF and UMF, with the FOU representing the region between them.

2.1.2 General type-2 fuzzy sets

A General Type-2 fuzzy set is defined as

$$\tilde{A} = \left\{(x,u), \, \mu_{\tilde{A}}(x,u) \mid x \in X, \; u \in U\right\}, \tag{2.10}$$

The SMF, denoted by $\mu_{\tilde{A}(x)}$, is the key element of GT2-FSs differing from the IT2-FSs. The next chapter will deeply investigate how the SMF shape can be represented in different settings [15, 17].

2.2 T2-FLSs: A Brief Overview

After defining T2-FSs (i.e., IT2-FS and GT2-FS), we will briefly introduce IT2-FLSs and GT2-FLSs. A T2-FLS consists of a fuzzification layer, a fuzzy rule base, an inference engine, a type reducer, and a defuzzification layer, as shown in Figure 2.1.

T2-FLSs are constructed based on if-then rule structures. T2-FSs are used to define the antecedent and consequent parts of T2-FLSs. First, the fuzzification layer is used

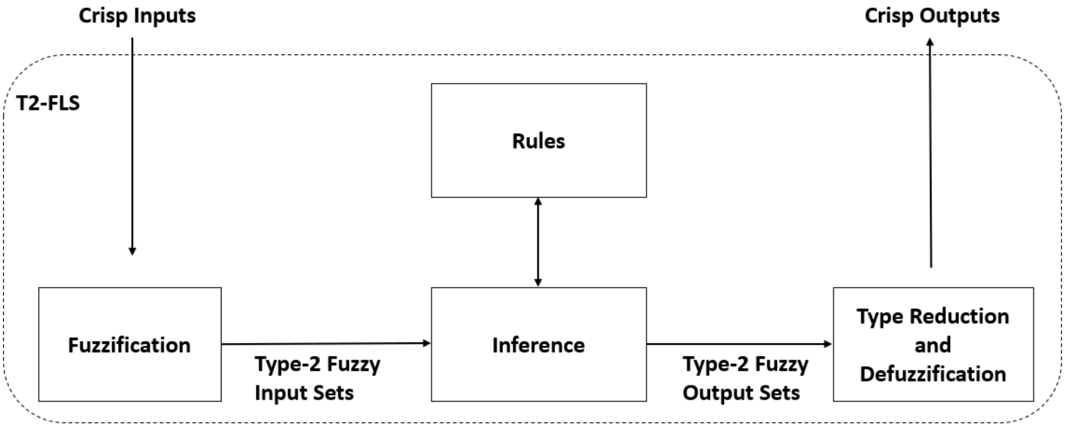


Figure 2.1 : Structure of a T2-FLS.

to transform crisp inputs into the T2 fuzzy sets [33], as shown in Figure 2.1. This process provides us with fuzzified inputs, subsequently, these fuzzy inputs are fed to the inference engine, which is responsible for applying the fuzzy rules in the rule base to derive the corresponding T2 fuzzy output sets [33]. These T2-FSs are reduced to T1-FSs via a type-reducer, and then the output of a T2-FLS is calculated by a defuzzification layer, as depicted in Figure 2.1 [33]. In this thesis study, we have used the Takagi-Sugeno-Kang type FLS as the T2-FLS. The antecedent membership functions of T2-FLSs are defined with T2-FSs, while the consequent membership functions of T2-FLSs are defined through affine (linear) functions. In the following subsections, we will introduce IT2-FLSs and GT2-FLSs, respectively.

2.2.1 Interval type-2 fuzzy logic systems

In this study, IT2-FLSs are constructed with gaussian antecedent functions and affine (linear) consequent membership functions. In this context, for a given input vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}}$ with M dimension and single input y, the rule structure of an IT2-FLS with P rules $(p = 1, 2, \ldots, P)$ is defined as follows:

$$R_p : \text{If } x_1 \text{ is } \tilde{A}_{p,1} \text{ and} \ldots x_M \text{ is } \tilde{A}_{p,M} \text{ Then } y \text{ is } y_p \tag{2.11}$$

where $\tilde{A}_{p,1}, \ldots, \tilde{A}_{p,M}$ are antecedent membership functions of the given input vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}}$ respectively, y is the output of the IT2-FLS. Here, R_p is the rule number, p represents the rule index. y_p is the consequent membership function, which is defined via an affine (linear) function as follows:

$$y_p = \sum_{m=1}^{M} a_{p,m} x_m + a_{p,0} \tag{2.12}$$

The antecedent MFs $\tilde{A}_{p,m}$ are defined with IT2-FSs that are represented via an Upper MF and a Lower MF as follows:

$$\overline{\mu}_{\tilde{A}_{p,m}}(x_m) = \exp\left(-\left(x_m - c_{p,m}\right)^2/2\overline{\sigma}^2_{p,m}\right) \tag{2.13}$$

$$\underline{\mu}_{\tilde{A}_{p,m}}(x_m) = h_{p,m}\exp\left(-\left(x_m - c_{p,m}\right)^2/2\underline{\sigma}^2_{p,m}\right) \tag{2.14}$$

where $c_{p,m}$ is the center, $\tilde{\sigma}_{p,m} = [\underline{\sigma}_{p,m}, \overline{\sigma}_{p,m}]$ is the standard deviation while $h_{p,m}$ defines the height of the LMF $\forall p,m$. The output of the IT2-FLS (y) is defined with the Type-Reduced Set $\tilde{Y} = [\underline{y}, \overline{y}]$ of the IT2-FLS and can be calculated as follows[17]:

$$y(\mathbf{x}) = (\underline{y}(\mathbf{x}) + \overline{y}(\mathbf{x}))/2 \tag{2.15}$$

where $\underline{y}$, and $\overline{y}$ represent the left and right boundary points of the type-reduced set, and these boundary points are obtained via a Center of Sets Calculation Method (CSCM) method. In this thesis study, we have used the Karnik-Mendel (KM) algorithm to calculate the type-reduced set as follows[14]:

$$\underline{y}(\mathbf{x}) = \frac{\sum_{p=1}^{L}\overline{f}_p(\mathbf{x})y_p + \sum_{p=L+1}^{P}\underline{f}_p(\mathbf{x})y_p}{\sum_{p=1}^{L}\overline{f}_p(\mathbf{x}) + \sum_{p=L+1}^{x}\underline{f}_p(\mathbf{x})} \tag{2.16}$$

$$\overline{y}(\mathbf{x}) = \frac{\sum_{p=1}^{R}\underline{f}_p(\mathbf{x})y_p + \sum_{p=R+1}^{P}\overline{f}_p(\mathbf{x})y_p}{\sum_{p=1}^{R}\underline{f}_p(\mathbf{x}) + \sum_{p=R+1}^{P}\overline{f}_p(\mathbf{x})} \tag{2.17}$$

where L and R represent the switching points of KM algorithm [14]. In (2.16) and (2.17), $\underline{f}_p(\mathbf{x})$ and $\overline{f}_p(\mathbf{x})$ represent the lower and upper firing strengths of the p^{th} rule and calculated as follows:

$$\underline{f}_p(\mathbf{x}) = \underline{\mu}_{\tilde{A}_{p,1}}(x_1) \cap \underline{\mu}_{\tilde{A}_{p,2}}(x_2) \cap \ldots \cap \underline{\mu}_{\tilde{A}_{p,M}}(x_M) \tag{2.18}$$

$$\overline{f}_p(\mathbf{x}) = \overline{\mu}_{\tilde{A}_{p,1}}(x_1) \cap \overline{\mu}_{\tilde{A}_{p,2}}(x_2) \cap \ldots \cap \overline{\mu}_{\tilde{A}_{p,M}}(x_M) \tag{2.19}$$

The membership grades of the LMFs are given by:

$$\underline{\mu}_{\tilde{A}_{p,1}},\ \underline{\mu}_{\tilde{A}_{p,2}},\ \ldots,\ \underline{\mu}_{\tilde{A}_{p,M}}, \quad \text{for } \mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}} \tag{2.20}$$

while the membership grades of the UMFs are:

$$\overline{\mu}_{\tilde{A}_{p,1}},\ \overline{\mu}_{\tilde{A}_{p,2}},\ \ldots,\ \overline{\mu}_{\tilde{A}_{p,M}}, \quad \text{for } \mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}} \tag{2.21}$$

Here, $\cap$ is the t-norm operator, which can be defined using various mathematical functions. Among the most commonly used t-norms are the minimum and product operators [14, 28, 34].

In this study, we used the product operator as the t-norm operator. During the statistical analysis of IT2-FLSs on various datasets, we used two different antecedent membership functions to define the IT2-FSs to construct parametric IT2-FLSs as [29]:

- H type IT2-FSs: As shown in Figure 2.2, the FOU is only generated by the $h_{p,m}$ $\forall p,m$, since $\underline{\sigma}_{p,m} = \overline{\sigma}_{p,m}$.
- HS type IT2-FSs: As shown in Figure 2.3, the FOU is dependent on the $h_{p,m}$, $\underline{\sigma}_{p,m}$, and $\overline{\sigma}_{p,m}, \forall p,m$.

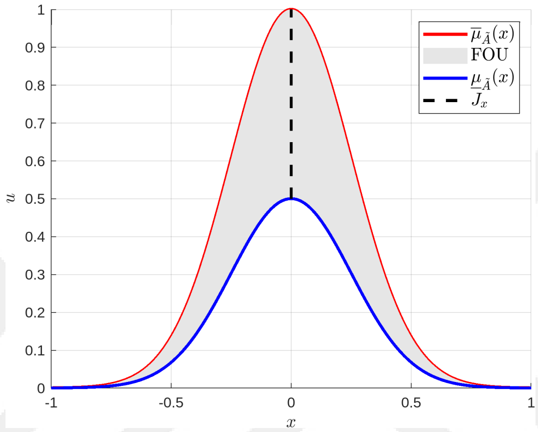


Figure 2.2 : H-IT2-FS.

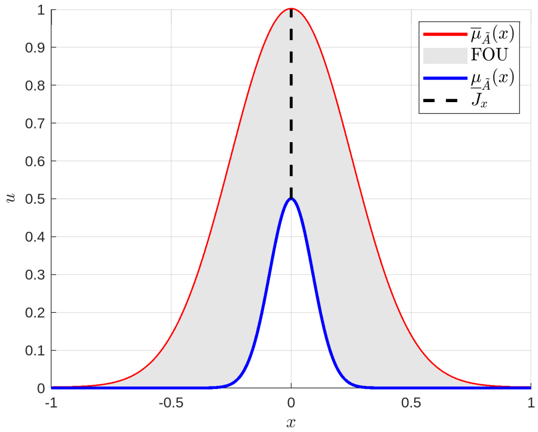


Figure 2.3 : HS-IT2-FS.

2.2.2 General type-2 fuzzy logic systems

GT2-FLSs are constructed based on GT2-FSs, which were first introduced by Zadeh [15, 16] and have since been widely studied, particularly following the definition provided by Mendel and John [17]. In this study, we differentiate between two types of GT2-FLSs based on their underlying GT2-FS definitions. Specifically, we refer to GT2-FLSs that utilize Zadeh's definition [15, 16] as Z-GT2-FLS, while those based on the definition by Mendel and John [17] are termed MJ-GT2-FLS. In this section we will give brief introduction on GT2-FLSs, then deep dive into Z/MJ-GT2-FLSs in the following chapter.

The GT2-FLS is formulated for an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}}$ and a single output y. The rule base is composed of P rules $(p = 1, 2, \ldots, P)$ that is defined as:

$$R_p : \text{If } x_1 \text{ is } \tilde{A}_{p,1} \text{ and} \ldots x_M \text{ is } \tilde{A}_{p,M} \text{ Then } y \text{ is } y_p \tag{2.22}$$

where $\tilde{A}_{p,1}, \ldots, \tilde{A}_{p,M}$ are antecedent membership functions of the given input vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}}$ respectively, y is the output of the GT2-FLS. Here, R_p is the rule number, p represents the rule index. y_p is the consequent membership function, which is defined via an affine (linear) function as follows:

$$y_p = \sum_{m=1}^{M} a_{p,m} x_m + a_{p,0} \tag{2.23}$$

The antecedent MFs are defined with GT2-FSs $\tilde{A}_{p,m}$ that are defined as a collection of α-planes (α_k) as follows:

$$\tilde{A}_{p,m} = \bigcup_{\alpha_k \in [0,1]} \tilde{A}_{p,m}^{\alpha_k} \tag{2.24}$$

where $\tilde{A}_{p,m}^{\alpha_k}$ is the α-plane of $\tilde{A}_{p,m}$ associated with $\alpha_k \in [0, 1]$. When α_k is distributed uniformly, we express it as $\alpha_k = k/K$ for k ranging from 0 to K. Thus, there are a total of $K + 1$ α-planes [14]. This representation allows defining the output of the GT2-FLSs as follows:

$$y(\boldsymbol{x}) = \frac{\sum_{k=0}^{K} y^{\alpha_k}(\boldsymbol{x}) \alpha_k}{\sum_{k=0}^{K} \alpha_k} \tag{2.25}$$

Here, $\mathrm{y}^{\alpha_k}(\boldsymbol{x})$ is the α_k.-IT2-FLS output and defined as:

$$y^{\alpha_k}(\boldsymbol{x}) = (\underline{y}^{\alpha_k}(\boldsymbol{x}) + \overline{y}^{\alpha_k}(\boldsymbol{x}))/2 \tag{2.26}$$

Here, $\underline{y}^{\alpha_k}$, and $\overline{y}^{\alpha_k}$ represent the left and right boundary points of the type-reduced set for the given α_k-IT2-FLS, and these boundary points are obtained via KM algorithm as follows[14]:

$$\underline{y}^{\alpha_k}(\boldsymbol{x}) = \frac{\sum_{p=1}^{L} \underline{f}_p^{\alpha_k}(\boldsymbol{x}) y_p + \sum_{p=L+1}^{P} \overline{f}_p^{\alpha_k}(\boldsymbol{x}) y_p}{\sum_{p=1}^{L} \underline{f}_p^{\alpha_k}(\boldsymbol{x}) + \sum_{p=L+1}^{P} \overline{f}_p^{\alpha_k}(\boldsymbol{x})} \tag{2.27}$$

$$\overline{y}^{\alpha_k}(\boldsymbol{x}) = \frac{\sum_{p=1}^{R} \underline{f}_p^{\alpha_k}(\boldsymbol{x}) y_p + \sum_{p=R+1}^{P} \overline{f}_p^{\alpha_k}(\boldsymbol{x}) y_p}{\sum_{p=1}^{R} \underline{f}_p^{\alpha_k}(\boldsymbol{x}) + \sum_{p=R+1}^{P} \overline{f}_p^{\alpha_k}(\boldsymbol{x})} \tag{2.28}$$

where L, R are the switching points of the KM algorithm [14]. $\underline{f}_p^{\alpha_k}(\boldsymbol{x})$ and $\overline{f}_p^{\alpha_k}(\boldsymbol{x})$ are the lower and upper rule firing of the p^{th} rule and are defined as:

$$\underline{f}_p^{\alpha_k}(\boldsymbol{x}) = \underline{\mu}_{\tilde{A}_{p,1}^{\alpha_k}}(x_1) \cap \underline{\mu}_{\tilde{A}_{p,2}^{\alpha_k}}(x_2) \cap \ldots \cap \underline{\mu}_{\tilde{A}_{p,M}^{\alpha_k}}(x_M) \tag{2.29}$$

$$\overline{f}_p^{\alpha_k}(\boldsymbol{x}) = \overline{\mu}_{\tilde{A}_{p,1}^{\alpha_k}}(x_1) \cap \overline{\mu}_{\tilde{A}_{p,2}^{\alpha_k}}(x_2) \cap \ldots \cap \overline{\mu}_{\tilde{A}_{p,M}^{\alpha_k}}(x_M) \tag{2.30}$$

The membership grades of the LMFs for a given α_k-IT2-FLS are given by:

$$\underline{\mu}_{\tilde{A}_{p,1}^{\alpha_k}}, \ldots, \underline{\mu}_{\tilde{A}_{p,M}^{\alpha_k}}, \quad \text{for } \mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}} \tag{2.31}$$

while the membership grades of UMFs for a given α_k-IT2-FLS are:

$$\overline{\mu}_{\tilde{A}_{p,1}^{\alpha_k}}, \ldots, \overline{\mu}_{\tilde{A}_{p,M}^{\alpha_k}}, \quad \text{for } \mathbf{x} = (x_1, x_2, \ldots, x_M)^{\mathrm{T}} \tag{2.32}$$

Here, $\cap$ is the t-norm operator, which can be defined using various mathematical functions. Among the most commonly used t-norms are the minimum and product operators [14, 28, 34]. In this study, we used the product operator as the t-norm operator.

2.2.3 The implementation of karnik-mendel algorithm

There are multiple type-reduction methods. In this thesis study, we implement the KM Algorithm [14]. This algorithm includes an iterative method to determine the left endpoint $(\underline{y}, \underline{y}^{\alpha_k})$ and the right endpoint $(\overline{y}, \overline{y}^{\alpha_k})$. L and R are found iteratively by checking which combination of upper and lower firing strengths (e.g., $\overline{f}_p^{\alpha_k}, \underline{f}_p^{\alpha_k}$) yields a consistent centroid calculation. This iterative process increases the inference time and complexity of IT2 / GT2-FLSs. To handle this issue, the KM algorithm can be thought of as a linear functional programming problem [35]. It can be observed that (2.16) and (2.17) (i.e., (2.27) and (2.28)) can be reformulated via $u_p \in 0, 1$ which defines an equivalent $f_p = \overline{f}_p u_p + \underline{f}_p(1 - u_p)$. However, this approach still requires an iterative process.

In our paper [36], we propose an efficient method to handle this problem by eliminating the needed optimization problems in (2.16) and (2.17) by evaluating

$$Y(\boldsymbol{u}) = X(\boldsymbol{u}) \oslash Z(\boldsymbol{u}) \tag{2.33}$$

where,

$$X(\boldsymbol{u}) = c_0 + c\boldsymbol{u}: \quad \alpha_0 = \sum\nolimits_p^P y_p \underline{f}_p(\mathbf{x})\ ;\ c_p = y_p(\overline{f}_p(\mathbf{x}) - \underline{f}_p(\mathbf{x}))\ ,\ p = 1, ..., P \tag{2.34}$$

where y_p represents the consequent part of the p^{th} rule, and $\underline{f}_p(\mathbf{x})$, and $\overline{f}_p(\mathbf{x})$ are the lower and upper firing strengths of the p^{th} rule.

$$Z(\boldsymbol{u}) = \beta_0 + \beta\boldsymbol{u} : \quad \beta_0 = \sum\nolimits_p^P \underline{f}_p \;;\; \beta_p = (\overline{f}_p(\mathbf{x}) - \underline{f}_p(\mathbf{x})) \;,\; p = 1, ..., P \tag{2.35}$$

with $\boldsymbol{u} \in \mathbb{R}^{P \times 2^P}$ that defines all binary combinations u_p as follows:

$$\boldsymbol{u} = \prod_{p=1}^{P} \{0, 1\}. \tag{2.36}$$

Here, $\prod$ represents the cartesian product. For instance, $\boldsymbol{u}$ for $P = 2$ is as follows:

$$\boldsymbol{u} = \prod_{p=1}^{2} \{0, 1\} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{2.37}$$

As $Y(\boldsymbol{u})$ includes all possible solutions, we can obtain $\underline{y}$ and $\overline{y}$:

$$\underline{y} = \min(Y(\boldsymbol{u})) \;,\; \overline{y} = \max(Y(\boldsymbol{u})) \tag{2.38}$$

This approach is introduced for IT2-FLSs and can be directly extended to GT2-FLSs via α-plane representation as follows:

$$Y^{\alpha_k}(\boldsymbol{u}) = X^{\alpha_k}(\boldsymbol{u}) \oslash Z^{\alpha_k}(\boldsymbol{u}) \tag{2.39}$$

where,

$$X^{\alpha_k}(\boldsymbol{u}) = c_0^{\alpha_k} + c^{\alpha_k}\boldsymbol{u} : \quad c_0^{\alpha_k} = \sum\nolimits_p^P y_p \underline{f}_p^{\alpha_k}(\mathbf{x}) \;;\; c_p^{\alpha_k} = y_p(\overline{f}_p^{\alpha_k}(\mathbf{x}) - \underline{f}_p^{\alpha_k}(\mathbf{x})) \;,\; p = 1, ..., P \tag{2.40}$$

where y_p represents the consequent part of the p^{th} rule, and $\underline{f}_p^{\alpha_k}(\mathbf{x})$, and $\overline{f}_p^{\alpha_k}(\mathbf{x})$ are the lower and upper firing strengths of the p^{th} rule for the given α_k-plane.

$$Z^{\alpha_k}(\boldsymbol{u}) = \beta_0^{\alpha_k} + \beta^{\alpha_k}\boldsymbol{u} : \quad \beta_0^{\alpha_k} = \sum\nolimits_p^P \underline{f}_p^{\alpha_k} \;;\; \beta_p^{\alpha_k} = (\overline{f}_p^{\alpha_k}(\mathbf{x}) - \underline{f}_p^{\alpha_k}(\mathbf{x})) \;,\; p = 1, ..., P \tag{2.41}$$

with $\boldsymbol{u} \in \mathbb{R}^{P \times 2^P}$ that defines all binary combinations u_p as follows:

$$\boldsymbol{u} = \prod_{p=1}^{P} \{0, 1\}. \tag{2.42}$$

Here, $\prod$ represents the cartesian product. For instance, $\boldsymbol{u}$ for $P = 2$ is as follows:

$$\boldsymbol{u} = \prod_{p=1}^{2} \{0, 1\} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{2.43}$$

As $Y^{\alpha_k}(\boldsymbol{u})$ includes all possible solutions for α_k-IT2-FLS, we can obtain $\underline{y}^{\alpha_k}$ and $\overline{y}^{\alpha_k}$:

$$\underline{y}^{\alpha_k} = \min(Y^{\alpha_k}(\boldsymbol{u})) \;,\; \overline{y}^{\alpha_k} = \max(Y^{\alpha_k}(\boldsymbol{u})) \tag{2.44}$$

These implementations provide a superior baseline for the inference and the training of the T2-FLSs through batches/mini-batches using DL optimizers by eliminating the iterative process in the original KM algorithm[14].

3. Z-GT2-FLS FOR ENHANCED LEARNING

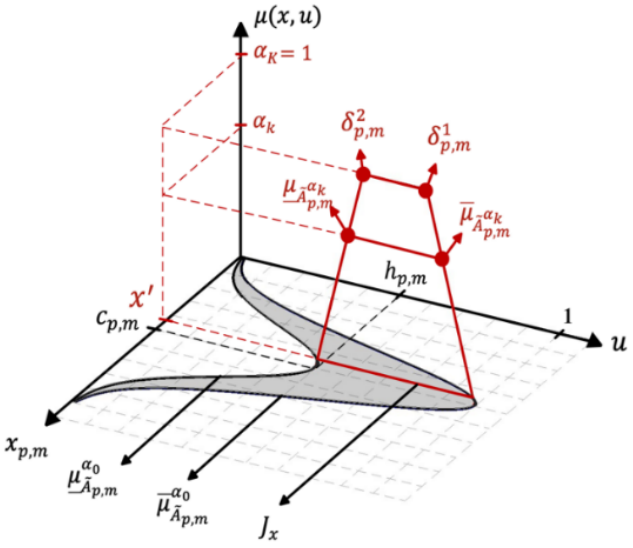
In this chapter, we will present the details of how to represent and learn high-performing GT2-FLSs. We begin by introducing MJ-GT2-FLSs [17] and discussing their potential limitations. Then, we propose Z-GT2-FLSs [15, 16] as a solution, highlighting how they effectively address the issues associated with MJ-GT2-FLSs.

3.1 MJ-GT2-FLS: Representation And Potential Issues

To define $\tilde{A}_{p,m}$, the most widely used GT2-FS representation is the one of Mendel & John [17] as shown in Figure 3.1. A GT2-FS $\tilde{A}$ is characterized by a type-2 MF $\big(x, \mu_{\tilde{A}}(x,u)\big)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\tilde{A} = \left\{ \big(x, \mu_{\tilde{A}}(x,u)\big) \mid x \in X, u \in J_x \subseteq [0,1] \right\} \tag{3.1}$$

in which $0 \le \mu_{\tilde{A}}(x,u) \le 1$ [17].



$\mu(x,u)$
$\alpha_K = 1$
α_k
$\delta^2_{p,m}$
$\delta^1_{p,m}$
$\underline{\mu}^{\alpha_k}_{\tilde{A}_{p,m}}$
$\overline{\mu}^{\alpha_k}_{\tilde{A}_{p,m}}$
$h_{p,m}$
x'
$c_{p,m}$
1
u
$x_{p,m}$
$\underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}$
$\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}$
J_x

Figure 3.1 : MJ-GT2-FS for $\underline{\sigma}_{p,m} = \overline{\sigma}_{p,m}$.

3.1.1 PMF representaton

According to the definition of MJ-GT2-FSs, we first parameterize a PMF, i.e. $\tilde{A}^{\alpha_0}_{p,m}$, by defining the following UMF $\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}(x_m)$ and LMF $\underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}(x_m)$:

$$\underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}(x_m) = h_{p,m} \exp\left(-\left(x_m - c_{p,m}\right)^2 / 2\underline{\sigma}^2_{p,m}\right) \tag{3.2}$$

$$\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}(x_m) = \exp\left(-\left(x_m - c_{p,m}\right)^2 / 2\overline{\sigma}^2_{p,m}\right) \tag{3.3}$$

where $c_{p,m}$ is the center, $\tilde{\sigma}_{p,m} = [\underline{\sigma}_{p,m}, \overline{\sigma}_{p,m}]$ is the standard deviation while $h_{p,m}$ defines the height of the LMF $\forall p, m$. As depicted in Figure 3.1, the Footprint of Uncertainty (FOU) (J_x) or the support of the Secondary MF (SMF) is defined with $\tilde{\sigma}_{p,m}$ and $h_{p,m}$.

3.1.2 SMF representation

In this study, we use the Trapezoid SMF, which employs a T1-FS which can be defined as follows:

$$\mu_{\tilde{A}(x)}(u) = \begin{cases} \frac{u-\delta^1}{\delta^2-\delta^1}, & \delta^1 \leq u < \delta^2 \\ 1, & \delta^2 \leq u < \delta^3 \\ \frac{\delta^4-u}{\delta^4-\delta^3}, & \delta^3 \leq u < \delta^4 \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Here, $\delta^1, \delta^2, \delta^3$ and δ^4 represent the left support, left core, right core, and right support points of the trapezoidal SMF, respectively. For a given α_k-IT2-FLS, LMF and UMF can be calculated as[37]:

$$\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}} = \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} + \left(\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}\right)\left(\delta^1_{p,m} + \alpha_k\left(\delta^2_{p,m} - \delta^1_{p,m}\right)\right) \tag{3.5}$$

$$\overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}} = \overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \left(\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}}\right)\left(1 - \delta^4_{p,m} + \alpha_k\left(\delta^4_{p,m} - \delta^3_{p,m}\right)\right) \tag{3.6}$$

Note that $0 \leq \delta^1 \leq \delta^2 \leq \delta^3 \leq \delta^4 \leq 1$.

In this study, we simplify the original parameter space by setting $\delta^1 = \underline{\mu}_{\tilde{A}_{p,m}}(x)$ and $\delta^4 = \overline{\mu}_{\tilde{A}_{p,m}}(x)$. Furthermore, for consistency and simplicity, we redefine the right core of the trapezoid as δ^1 and the left core as δ^2. With this reformulation, (3.4) is rewritten as follows:

$$\mu_{\tilde{A}(x)}(u) = \begin{cases} \frac{u-\underline{\mu}_{\tilde{A}_{p,m}}(x)}{\delta^2-\underline{\mu}_{\tilde{A}_{p,m}}(x)}, & \underline{\mu}_{\tilde{A}_{p,m}}(x) \leq u < \delta^2 \\ 1, & \delta^2 \leq u < \delta^1 \\ \frac{\overline{\mu}_{\tilde{A}_{p,m}}(x)-u}{\overline{\mu}_{\tilde{A}_{p,m}}(x)-\delta^1}, & \delta^1 \leq u < \overline{\mu}_{\tilde{A}_{p,m}}(x) \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

Based on the PMF, the following parameterized SMF is used that is defined via the UMF and LMF of $\tilde{A}^{\alpha_k}_{p,m}(k \neq 0)$ [37]:

$$\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}} = \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} + \alpha_k \left(\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} \right) \delta^1_{p,m} \tag{3.8}$$

$$\overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}} = \overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \alpha_k \left(\overline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} - \underline{\mu}^{\alpha_0}_{\tilde{A}_{p,m}} \right) \left(1 - \delta^2_{p,m} \right) \tag{3.9}$$

Here, $\left\{ \delta^1_{p,m}, \delta^2_{p,m} : \delta^1_{p,m} \geq \delta^2_{p,m} \right\} \in [0,1], \forall p, m$ are parameters that define the shape of the SMFs as shown in Figure 3.1. By α-plane representation, we can find the lower and upper membership grades (i.e., $\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}$ and $\overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}$) for the corresponding α_k-IT2-FLS.

3.1.3 Potential issues

In a recent study [30], a DL-based learning method for MJ-GT2-FLS, which is based on IT2-FLS one in [29], is presented. We identified the following two problems:

1. **Flexibility:** The drawback of the method lies in its insistence on the explicit parameterization of SMFs with respect to the PMFs prior to training, i.e. the implementation of the MJ-GT2-FLS definition via (3.8) and (3.9). While providing structure, this might hinder the GT2-FLS's learning capacity as the learning performance depends on how the shapes of the UMF and LMF in (3.2) and (3.3) (i.e. the FOU size) are defined.

2. **Curse of dimensionality:** This problem is a well-known problem for FLSs because lower and upper firing strengths approach zero $[\underline{f}_p^{\alpha_k}(\boldsymbol{x}), \overline{f}_p^{\alpha_k}(\boldsymbol{x})] \rightarrow 0$ when handling high-dimensional datasets (i.e., rule firing problem). In [30], they handled this problem by setting the t-norm operator $\cap$ in (2.29) and (2.30) w.r.t the data size (N) and dimension (M). They suggested using the product operator for low dimensional input vector spaces while the min one for high dimensional ones based on their exhaustive comparative results.

3.2 Z-GT2-FLS: Representation And Solutions

We conceptually travel back to the original formulation of GT2-FSs introduced by Zadeh [15, 16], adopting his definition to enhance flexibility during the learning process. This approach offers a practical solution to the curse of dimensionality problem, which often arises in FLSs.

Let us start to define GT2-FSs with the definition of Zadeh [15, 16] i.e., a GT2-FS $\tilde{A}$ on X is a mapping [15]:

$$\tilde{A} : X \rightarrow \mathrm{FS}\left([0,1]\right) \tag{3.10}$$

Alternatively, we can state that $\tilde{A}_{p,m}$ is mapping $\tilde{A}_{p,m} : X \rightarrow [0,1]^{[0,1]}$. The equivalence of (3.1) and (3.10) is given in [38]. In this study, we adopted (3.10) and integrated it with α-plane representation to define the output of Z-GT2-FLS via (2.25).

3.2.1 PMF representation

As shown in Figure 3.2, we represent the PMF with a T1-FS $A_{p,m}$ that is defined as follows:

$$\mu_{A_{p,m}}(x_m) = \exp\left(-\left(x_m - c_{p,m}\right)^2 / 2(\sigma_{p,m})^2\right) \tag{3.11}$$

where $\sigma_{p,m}$ is the standard deviation and $c_{p,m}$ is the center of the Gaussian MF, which defines the membership grade of a T1-FS $A_{p,m}$.

3.2.2 SMF representation

In this thesis study, we define the SMF with a two-sided Gaussian MF as follows:

$$\mu_{x_{p,m}}(u) = \begin{cases} \exp\left(-\dfrac{(u-\gamma_{p,m})^2}{2(\sigma^l_{p,m})^2}\right), \text{if } u \leq \gamma_{p,m} \\ \exp\left(-\dfrac{(u-\gamma_{p,m})^2}{2(\sigma^r_{p,m})^2}\right), \text{if } u \geq \gamma_{p,m} \end{cases} \tag{3.12}$$

Here,$\sigma^l_{p,m}$ and $\sigma^r_{p,m}$ are the left and right standard deviations, and $\gamma_{p,m}$ is the center, which defines the shape and support of the SMF. As shown in Figure 3.2, we set $\gamma_{p,m} = \mu_{A_{p,m}}(x_m)$.

Now, to extract the α-planes of the Z-GT2-FS ($\tilde{A}^{\alpha_k}_{p,m}$), we define the α-cuts of $\mu_{x_{p,m}}(u)$. We first rewrite (3.12) as follows:

$$-\sqrt{-2\ln(\alpha_k)}\sigma^l_{p,m} = \left(u - \gamma_{p,m}\right), \text{if } u \leq \gamma_{p,m} \tag{3.13}$$

$$\sqrt{-2\ln(\alpha_k)}\sigma^r_{p,m} = \left(u - \gamma_{p,m}\right), \text{if } u \geq \gamma_{p,m} \tag{3.14}$$

Then, by inserting $\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}$ into (3.13) and (3.14) while $\overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}$ into (3.13) and (3.14) as u, we can extract the LMF and UMF of $\tilde{A}^{\alpha_k}_{p,m}(k \neq 0)$ as follows:

$$\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m) = \mu_{A_{p,m}}(x_m) - \sqrt{-2\ln(\alpha_k)}\sigma^l_{p,m} \tag{3.15}$$

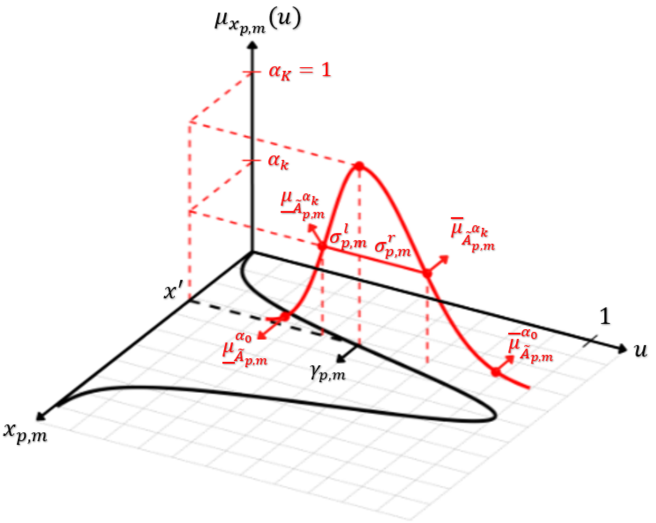

$\mu_{x_{p,m}}(u)$
$\alpha_K = 1$
α_k
$\underline{\mu}_{\tilde{A}_{p,m}}^{\alpha_k}$
$\sigma_{p,m}^{l}$
$\sigma_{p,m}^{r}$
$\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_k}$
x'
1
u
$\underline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$
$\gamma_{p,m}$
$\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$
$x_{p,m}$

Figure 3.2 : Z-GT2-FS.

$$\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_k}(x_m) = \mu_{A_{p,m}}(x_m) + \sqrt{-2\ln(\alpha_k)}\sigma_{p,m}^{r} \tag{3.16}$$

Now, by using (3.15) and (3.16), we can define the output of Z-GT2-FLS via α-plane representation as given in (2.25). The only problem with this implementation is due to the domain space of $\ln(\cdot)$, which spans $(0,\infty]$, and thus $\alpha_0 = 0$ is not included. Thus, we associate the α_0-plane with $\alpha_0 \triangleq 0.01$. Our motivation for this setting is grounded in the consideration that $\sqrt{-2\ln(\alpha_0)} \approx 3$. Thus, $\underline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$ and $\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$ are defined explicitly as follows:

$$\underline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}(x_m) \triangleq \mu_{A_{p,m}}(x_m) - 3\sigma_{p,m}^{l} \tag{3.17}$$

$$\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}(x_m) \triangleq \mu_{A_{p,m}}(x_m) + 3\sigma_{p,m}^{r} \tag{3.18}$$

Thus, we can ensure that $\underline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$ and $\overline{\mu}_{\tilde{A}_{p,m}}^{\alpha_0}$ remain within a range of three standard deviations from $\gamma_{p,m}$.

3.2.3 Curse of dimensionality problem

Unlike [30], our study adopts a consistent approach by using the product operator as $\cap$ in (2.30) regardless of the data size and dimension. To avoid the general problem of $[\underline{f}_p^{\alpha_k}(\boldsymbol{x}), \overline{f}_p^{\alpha_k}(\boldsymbol{x})] \rightarrow 0$ in high-dimensional datasets (i.e. rule firing problem), we propose a method like the HTSK [39] that scales $\mu_{A_{p,m}}$ w.r.t M as follows:

$$\mu_{A_{p,m}}^{*} = (\mu_{A_{p,m}})^{1/M} \tag{3.19}$$

We can reformulate (3.19) in the following explicit standard form of a Gaussian MF to represent the PMF:

$$\mu_{A_{p,m}}^{*}(x_m) = \exp\left(-\left(x_m - c_{p,m}\right)^2 / (2(\sqrt{M}\sigma_{p,m})^2)\right) \tag{3.20}$$

which is mathematically equivalent to HTSK [39] as shown in [40]. When compared to (3.11), we observe that $\sigma_{p,m}$ of the PMF is scaled with $\sqrt{M}$. Thus, the learning of GT2-FLS is not substantially affected by the increase in feature dimensionality.

3.3 Learning T2-FLSs Within DL Frameworks

We first define the Learnable Parameters (LPs) of the T2-FLSs and then provide parameterization tricks to allow learning them via DL optimizers and AD methods provided within DL frameworks such as Matlab and PyTorch.

3.3.1 Learnable parameter sets for IT2-FLSs

The LP sets of the IT2-FLSs ($\boldsymbol{\theta_{IT2}}$) consist of the LPs of antecedent MF ($\boldsymbol{\theta}_{IT2-A}$) and those of the consequent MFs ($\boldsymbol{\theta}_{IT2-C}$) [29]. The LP set of IT2 antecedent MF is:

- H type IT2-FS: the set is $(\boldsymbol{\theta}_{IT2-AH}) = \{\boldsymbol{c},\boldsymbol{\sigma},\boldsymbol{h}\}$, where $\boldsymbol{c} = (c_{1,1},\ldots,c_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{\sigma} = (\sigma_{1,1},\ldots,\sigma_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{h} = (h_{1,1},\ldots,h_{P,M})^T \in \mathbb{R}^{P\times M}$, $(\sigma_{p,m} = \underline{\sigma}_{p,m} = \overline{\sigma}_{p,m})$.
- HS type IT2-FS: the set is $(\boldsymbol{\theta}_{IT2-AHS}) = \{\boldsymbol{c},\underline{\boldsymbol{\sigma}},\overline{\boldsymbol{\sigma}},\boldsymbol{h}\}$, where $\boldsymbol{c} = (c_{1,1},\ldots,c_{P,M})^T \in \mathbb{R}^{P\times M}$, $\underline{\boldsymbol{\sigma}} = (\underline{\sigma}_{1,1},\ldots,\underline{\sigma}_{P,M})^T \in \mathbb{R}^{P\times M}$, $\overline{\boldsymbol{\sigma}} = (\overline{\sigma}_{1,1},\ldots,\overline{\sigma}_{P,M})^T \in \mathbb{R}^{P\times M}$,
$\boldsymbol{h} = (h_{1,1},\ldots,h_{P,M})^T \in \mathbb{R}^{P\times M}$.

The learnable parameter set of consequent MF is identical, $\boldsymbol{\theta}_{\mathrm{IT2-C}} = \{\boldsymbol{a},\boldsymbol{a}_0\}$, with $\boldsymbol{a} = (\boldsymbol{a}_{1,1},\ldots,\boldsymbol{a}_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{a}_0 = (a_{1,0},\ldots,a_{P,0})^T \in \mathbb{R}^{P\times 1}$. Overall, H-IT2-FLS has a total of $3PM + P(M+1)$, while the HS-IT2-FLS includes $4PM + P(M+1)$ LPs.

3.3.2 Learnable parameter sets for GT2-FLSs

The LP sets of the GT2-FLSs ($\boldsymbol{\theta_{GT2}}$) consist of the LPs of antecedent MF ($\boldsymbol{\theta}_{GT2-A}$) and those of the consequent MFs ($\boldsymbol{\theta}_{GT2-C}$). For both the MJ-GT2-FLS and Z-GT2-FLS, we define the identical $\boldsymbol{\theta}_{GT2-C}$ as $\boldsymbol{\theta}_{\mathrm{GT2-C}} = \{\boldsymbol{a},\boldsymbol{a}_0\}$, with $\boldsymbol{a} = (\boldsymbol{a}_{1,1},\ldots,\boldsymbol{a}_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{a}_0 = (a_{1,0},\ldots,a_{P,0})^T \in \mathbb{R}^{P\times 1}$. The only difference between them lies in $\boldsymbol{\theta}_{\mathrm{GT2-A}} = \{\boldsymbol{\theta}_{\mathrm{AP}},\boldsymbol{\theta}_{\mathrm{AS}}\}$, i.e., (3.1) vs. (3.10).

- For MJ-GT2-FLS: $\boldsymbol{\theta}_{\mathrm{AP}} = \{\boldsymbol{c},\boldsymbol{\sigma},\boldsymbol{h}\}$ and $\boldsymbol{\theta}_{\mathrm{AS}} = \{\boldsymbol{\delta}^{(1)},\boldsymbol{\delta}^{(2)}\}$ with $\boldsymbol{c} = (c_{1,1},\ldots,c_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{\sigma} = (\sigma_{1,1},\ldots,\sigma_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{h} = (h_{1,1},\ldots,h_{P,M})^T \in \mathbb{R}^{P\times M}$, $\boldsymbol{\delta}^{(1)} = (\delta_1^{(1)},\ldots,\delta_M^{(1)})^T \in \mathbb{R}^{1\times M}$, and $\boldsymbol{\delta}^{(2)} = (\delta_1^{(2)},\ldots,\delta_M^{(2)})^T \in \mathbb{R}^{1\times M}$. In [30], they set $\delta_{p,m}^1 = \delta_m^1$ and $\delta_{p,m}^2 = \delta_m^2, \forall p$.
- For Z-GT2-FLS, $\boldsymbol{\theta}_{\mathrm{AP}} = \{\boldsymbol{c},\boldsymbol{\sigma}\}$ and $\boldsymbol{\theta}_{\mathrm{AS}} = \{\boldsymbol{\sigma^l},\boldsymbol{\sigma^r}\}$ with $\boldsymbol{\sigma}^{(l)} = (\sigma_1^{(l)},\ldots,\sigma_M^{(l)})^T \in$

$\mathbb{R}^{1\times M}$, and $\boldsymbol{\sigma}^r = (\sigma_1^{(r)}, \dots, \sigma_M^{(r)})^T \in \mathbb{R}^{1\times M}$. For the sake of simplicity, we set $\sigma_{p,m}^l = \sigma_m^l$ and $\sigma_{p,m}^r = \sigma_m^r, \forall p$.

To sum up, MJ-GT2-FLS has a total of $(3P+2)M + P(M+1)$, while the Z-GT2-FLS involves $(2P+2)M + P(M+1)$ LPs. Despite the added complexity of using a Gaussian SMF, the Z-GT2-FLS has PM fewer LPs compared to MJ-GT2-FLS.

3.3.3 Parameterization tricks for T2-FLSs for DL optimizers

The learning problem of GT2-FLSs is defined with constraints $\boldsymbol{\theta} \in \boldsymbol{C}$ that arise from the definitions of FSs [29]. Given that DL optimizers are unconstrained ones, we introduce parametrization tricks to transform $\boldsymbol{\theta}$ to an unbounded search space. In this context, we will present the parameterization tricks for both IT2-FLSs and MJ-GT2-FLSs [29, 30]. Afterwards, we will introduce the parameterization tricks for Z-GT2-FLSs.

- Parameterization Tricks for IT2-FLSs: For the antecedent part, we must satisfy the condition $\overline{\mu}_{A_{p,m}}(x_m) \geq \underline{\mu}_{A_{p,m}}(x_m)$ [29]. Thus, we have the constraints:

$$0 \leq h_{p,m} \leq 1, \ \underline{\sigma}_{p,m} \leq \overline{\sigma}_{p,m} \tag{3.21}$$

  To solve this issue, we implement the following parameterization tricks :

$$h_{r,m} = \mathrm{sig}(h'_{r,m}) \tag{3.22}$$

$$\overline{\sigma}_{p,m} = \sigma'_{p,m} + |\Delta|, \quad \underline{\sigma}_{p,m} = \sigma'_{p,m} - |\Delta| \tag{3.23}$$

  where $sig(\cdot)$ is the sigmoid function and $\{h'_{r,m}, \sigma'_{p,m}, \Delta\}$ are the new unbounded learnable parameters [29].

- Parameterization Tricks for MJ-GT2-FLSs: For the antecedent part, (3.22) is implemented. For the SMF part, we have the following constraint [30]:

$$0 \leq \delta_m^2 \leq \delta_m^1 \leq 1. \tag{3.24}$$

  Thus, to satisfy the condition, we utilize the following parameterization tricks[30]:

$$\delta_m^1 = sig(\delta_m^{1\,\prime}) \quad \delta_m^2 = \delta_m^1 sig(\delta_m^{2\,\prime}) \tag{3.25}$$

  where $sig(\cdot)$ is the sigmoid function and $\{\delta_m^{1\,\prime}, \delta_m^{2\,\prime}\}$ are the new unbounded learnable parameters [30].

- Parameterization Tricks for Z-GT2-FLSs: For $\boldsymbol{\sigma^l}, \boldsymbol{\sigma^r} \in \boldsymbol{\theta}_{GT2-A}$, we must ensure that the learned $\tilde{A}^{\alpha_k}_{p,m}$ adhere to the conditions of GT2-FSs, specifically $0 \leq \underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m) \leq \bar{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m) \leq 1, \forall p, m$. It is important to highlight that, as per the definitions in (3.15) and (3.16), we inherently ensure $\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m) \leq \overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m), \forall p, m$.

  For $\boldsymbol{\sigma^l}$, we address the constraint $0 \leq \sigma^l_m \leq \gamma_{p,m}/\sqrt{-2\ln(0.01)}$ via:

$$\sigma^l_m = \gamma_{p,m}/\sqrt{-2\ln(0.01)}\,\mathrm{sig}(\widehat{\sigma}^l_m) \tag{3.26}$$

  For $\boldsymbol{\sigma^r}$, where the constraint is $0 \leq \sigma^r_m \leq \left(1-\gamma_{p,m}\right)/\sqrt{-2\ln(0.01)}$, we do the following trick:

$$\sigma^r_m = \left(1-\gamma_{p,m}\right)/\sqrt{-2\ln(0.01)}\,\mathrm{sig}(\widehat{\sigma}^r_m) \tag{3.27}$$

  where $\mathrm{sig}(\cdot)$ is the sigmoid function that provides the generation of unbounded optimization variables $\{\widehat{\sigma}^l_m, \widehat{\sigma}^r_m\}$. Here, the utilization of $1/\sqrt{-2\ln(0.01)}$ is motivated by the objective to ensure $[\underline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m), \overline{\mu}_{\tilde{A}^{\alpha_k}_{p,m}}(x_m)] \in [0, 1], \forall p, m$.

4. LEARNING T2-FLSs WITH A DUAL-FOCUS

In this chapter, we introduce the proposed framework for learning dual-focused Type-2 Fuzzy T2-FLSs and provide a comprehensive analysis of our Z-GT2-FLSs model. We demonstrate in detail how it outperforms existing models in terms of both point-wise estimation and generating HQ-PIs[9].

4.1 The Learning Framework For Accuracy & HQ-PI

Here, we introduce a DL framework designed to enable the learning of T2-FLSs that not only yield accurate point-wise predictions but also excel in generating HQ-PI characterized by high uncertainty coverage with tight PI bands. The Table 4.1 provides the algorithm for training steps of dual-focused GT2-FLS for a dataset $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, where $\boldsymbol{x}_n = \left(x_{n,1}, \ldots, x_{n,M}\right)^T$. As we aim to learn a dual-focused IT2/GT2-FLS, we defined the following loss to be minimized by a DL optimizer [29]:

$$\min_{\boldsymbol{\theta} \in C} L = \frac{1}{N} \sum_{n=1}^{N} \left(L_R\left(x_n, y_n\right) + \ell\left(x_n, y_n, \underline{\tau}, \overline{\tau}\right) \right) \tag{4.1}$$

Here, the constraints $\boldsymbol{C}$ are eliminated as described in Section 3.3.3. The loss function has an accuracy-focused part $L_R(\cdot)$ and an uncertainty-focused part $\ell(\cdot)$.

We first review the composite loss function definition for IT2-FLSs[29] then, we will describe how this loss definition can be extended to GT2-FLSs in two different ways [30, 41].

4.1.1 Composite loss definition for IT2-FLSs

Here, the LP set ($\boldsymbol{\theta_{IT2}}$) is used in (4.1). $L_R(\cdot)$ represents the empirical risk function for the accuracy part and is defined as follows [29]:

$$L_R(\epsilon_n) = \log(\cosh(\epsilon_n)) \tag{4.2}$$

Here ϵ_n is the point-wise accuracy error and is defined as:

$$\epsilon_n = y_n - y_{IT2}\left(x_n\right) \tag{4.3}$$

Table 4.1 : DL-based Dual-Focused GT2-FLS Training Algorithm.

Step	Description
1	**Input:** N training samples $(x_n, y_n)_{n=1}^{N}$, $\phi = [\underline{\tau}, \overline{\tau}]$
2	Set $K+1$: number of α-planes
3	Set P: number of rules
4	Set mbs: mini-batch size
5	Set T: number of epochs
6	**Output:** Learned parameter set θ_{GT2}
7	Initialize $\theta_{GT2} = [\theta_{GT2-A}, \theta_{GT2-C}]$
8	**For** $t = 1$ to T **do**
9	**For each** mbs in N **do**
10	Perform parametrization tricks for θ (see Section 3.3.3)
11	$\mu^* \leftarrow \mathrm{PMF}(x; \theta_{AP})$ (3.11)
12	$[\underline{\mu}^{\alpha_k}, \overline{\mu}^{\alpha_k}] \leftarrow \mathrm{SMF}(\mu^*; \theta_{AS})$ (3.15) and (3.16)
13	$[\underline{y}^{\alpha_0}, \overline{y}^{\alpha_0}, y] \leftarrow \mathrm{Inference}(\underline{\mu}^{\alpha_k}, \overline{\mu}^{\alpha_k}; \theta_C)$ (see Section 2.2)
14	Compute L_1 or L_2 (4.9) or (4.10)
15	Compute $\partial L / \partial \theta$ via automatic differentiation (AD)
16	Update θ via a DL optimizer, e.g., Adam
17	**End for**
18	**End for**
19	$\theta^* \leftarrow \arg\min L$
20	**Return:** θ^*

where $y_{IT2}(x_n)$ represents the defuzzified output of the IT2-FLS as in (2.15). For the uncertainty-focused part, $\ell(\cdot)$ is constructed via a pinball loss $\rho(\cdot)$ that is defined as:

$$\rho(x_n, y_n, y, \tau) = \max(\tau(y_n - y(x_n)), (\tau - 1)(y_n - y(x_n))) \tag{4.4}$$

Here τ defines the desired quantile level to be covered. For learning an envelope that captures the expected amount of uncertainty, we define a lower ($\underline{\tau}$) and upper ($\overline{\tau}$) quantile level. We utilize TR set of the IT2-FLS, $[\underline{y}(x_n), \overline{y}(x_n)]$ as our lower and upper bound predictions and define the following loss:

$$\ell\left(x_n, y_n, \underline{y}, \overline{y}, \underline{\tau}, \overline{\tau}\right) = \rho(x_n, y_n, \underline{y}, \underline{\tau}) + \rho\left(x_n, y_n, \overline{y}, \overline{\tau}\right) \tag{4.5}$$

4.1.2 Composite loss function for GT2-FLSs

Here, we utilize the LP set ($\boldsymbol{\theta_{GT2}}$) in (4.1). Thanks to the structure of GT2-FSs via α- plane representation, we can define two loss definitions. First, we utilize TR set of α_0-plane, $[\underline{y}^{\alpha_0}(x_n), \overline{y}^{\alpha_0}(x_n)]$ as our lower and upper bound predictions and define the following loss:

$$\ell\left(x_n, y_n, \underline{y}, \overline{y}, \underline{\tau}, \overline{\tau}\right) = \rho\left(x_n, y_n, \underline{y}^{\alpha_0}, , \underline{\tau}\right) + \rho\left(x_n, y_n, \overline{y}^{\alpha_0}, \overline{\tau}\right) \tag{4.6}$$

For the accuracy part $L_R(\cdot)$, the error term (ϵ_n) can be defined via two different ways [30, 41]:

$$\epsilon_n = y_n - y_{GT2}(x_n) \tag{4.7}$$

where $y_{GT2}(x_n)$ is the aggregated output of the GT2-FLS as in (2.25). On the other hand, instead of using the aggregated output of the GT2-FLSs, we can use the defuzzified output of the α_K-IT2-FLS ($y^{\alpha_K}(x_n)$, where $\alpha_K = 1$) [30, 41]. So the new error term, $\epsilon_n^{\alpha_K}$ is as:

$$\epsilon_n^{\alpha_K} = y_n - y^{\alpha_K}(x_n) \tag{4.8}$$

To summarize, we define the following two loss functions (L_1 and L_2) for learning dual-focused GT2-FLS:

$$L_1 = \frac{1}{N}\sum_{n=1}^{N}\left[L_R(\epsilon_n) + \ell\left(x_n, y_n, \underline{y}, \overline{y}, \underline{\tau}, \overline{\tau}\right)\right] \tag{4.9}$$

where ϵ_n is given in (4.7) and

$$L_2 = \frac{1}{N}\sum_{n=1}^{N}\left[L_R(\epsilon_n^{\alpha_K}) + \ell\left(x_n, y_n, \underline{y}, \overline{y}, \underline{\tau}, \overline{\tau}\right)\right] \tag{4.10}$$

where $\epsilon_n^{\alpha_K}$ is given in (4.8). Through the loss functions, an (partially) independent learning of $\boldsymbol{\theta_{GT2}}$ is possible for UQ while achieving high accuracy.

4.2 Comparative Performance Analysis

We evaluate the performance of our Z-GT2-FLS against MJ-GT2-FLS [30] and two interval Type-2 FLSs [29] on five regression benchmarks: White Wine (WW), Powerplant (PP), Abalone (ABA), Parkinson Motor UPDRS (PM), and AIDS.

4.2.1 Design of experiments

All input features are standardized via Z-score normalization:

$$\tilde{x}_i = \frac{x_i - \mu_x}{\sigma_x} \tag{4.11}$$

where μ_x and σ_x are the sample mean and standard deviation of feature x over the full dataset. The target variable y is similarly normalized on the *training set*:

$$\tilde{y} = \frac{y - \mu_y}{\sigma_y} \tag{4.12}$$

where μ_y and σ_y denote the sample mean and standard deviation of y. After inference, predictions are mapped back via

$$y = \tilde{y}\,\sigma_y + \mu_y \tag{4.13}$$

Each dataset is randomly split into a training set (70%) and a test set (30%) by following [29, 30]. As a note, prior studies [29, 30] normalize using the entire dataset (both training and test sets). Therefore, for the purpose of consistent and meaningful benchmarking, we also apply Z-score normalization using the full dataset when comparing our models to others. All models (Z-GT2-FLS, MJ-GT2-FLS, and the two IT2-FLSs) are trained with identical hyperparameters:

- Mini-batch size: 64
- Learning rate: 1×10^{-3}
- Number of epochs: 100 (except PM dataset: 1000 epochs)

The target PI coverage level is set to 99%, i.e.,

$$\varphi = [0.005, 0.995]. \tag{4.14}$$

The experiments were conducted within MATLAB ® and repeated with 20 different initial seeds for statistical analysis. We configured the GT2-FLSs with 3 α-planes ($\alpha = [0.01, 0.5, 1]$) and trained them by utilizing L_1 defined in (4.9) (GT2-FLS-1) and L_2 defined in (4.10) (GT2-FLS-2). The two IT2-FLSs, namely IT2-FLS-H and IT2-FLS-HS, are defined with IT2-FSs, which we define in (2.9) and are learned via the DL-based approach presented in [29].In the experiments, we set $P = 5$ and $P = 10$ for all FLSs to analyze how the number of P affects the model performance. Note that, P is a hyperparameter of FLSs, and can be thought of as the number of neurons in the neural network architectures. The number of LP (#LP) of FLSs for the handled datasets is tabulated in Tables 4.3 and 4.4. It can be observed that the IT2-FLS-HS has the largest number of LPs.

4.2.2 Performance evaluation

We evaluated the performances via Root Mean Square Error (RMSE), PI Coverage Probability (PICP), and PI Normalized Averaged Width (PINAW) [10]. We anticipate training an FLS that yields a low RMSE, i.e., high accuracy and attains a PICP of 99% with a low PINAW, thereby indicating an HQ-PI as defined in [9].

Tables 4.3 and 4.4 provide the mean RMSE, PICP, and PINAW alongside their ±1 standard error values over 20 experiments for $P = 5$ and $P = 10$, respectively to understand how the rule size P affects the performance of dual-focused IT2 and GT2 FLSs. We also provided the rankings over the 5 datasets handled for an easy comparison. Furthermore, we include a comprehensive computational load analysis to comment on the training complexity in relation to P.

4.2.2.1 Performance analysis for $P = 5$

In this subsection, we evaluate the performance of T2-FLSs for $P = 5$. In Figures 4.1, 4.2, 4.3, 4.4 and 4.5, for statistical analysis, we present the notched box and whisker plots showing median (central mark), 25th/75th percentiles (left and right edges of box) which defines the Inter Quartile Range (IQR), whiskers (line), and outliers (circles). Observe that:

- For WW, the Z-GT2-FLS-1 resulted in the best RMSE performance with a statistically significant difference, due to the absence of notches overlapping, as shown in Figure 4.2. Z-GT2-FLS-1 excels slightly in accuracy, while Z-GT2-FLS-2 shows a marginally higher PICP value. IT2-FLS-HS stands out with the lowest PINAW value.

- For PP, all FLSs show competitive results in terms of accuracy, reliability, and precision. On the other hand, IT2-FLS-HS has slightly better RMSE and PICP values while IT2-FLS-H stands out lowest PINAW. As shown in Figure 4.5, there is no significant difference between FLSs.

- For ABA, MJ-GT2-FLS-2 stands out as the best one in terms of RMSE, closely followed by other models. Z-GT2-FLS-2 demonstrates the highest reliability in capturing true values within PI and has a low PINAW value (i.e. HQ-PI). Generally, as shown in Figure 4.4, all GT2-FLSs have similar results in performance metrics, yet are significantly better than their IT2 counterparts.

- For PM, The Z-GT2-FLS-2 demonstrates superior performance with the lowest RMSE which is statistically significant when compared to MJ-GT2 and IT2 counterparts. Z-GT2-FLS models have similar performances in terms of RMSE and PICP as shown in Figure 4.1. Z-GT2-FLS-2 has the lowest mean PINAW, indicating the generation HQ-PIs. IT2-FLS-HS also offers good coverage, but at the cost of wider PIs.

- For AIDS, the point-wise predictions of all FLSs are similar, yet the Z-GT2-FLS-1 has a better mean RMSE value. On the other hand, as shown in Figure 4.3, the IT2-FLS-HS has the best PICP, yet its PINAW is significantly larger than other FLS counterparts, indicating a poor PI. While Z-GT2-FLS-1 has a similar PI performance with a much narrower PI band (i.e. low PINAW) and thus an HQ-PI, Z-GT2-FLS-2 exhibits the lowest mean PINAW.

To sum up, Z-GT2-FLSs, with minimal #LPs, consistently demonstrate competitive performance across various datasets. Their capacity to precisely predict different target variables with HQ-PI establishes them as consistently robust performers. Yet, the Z-GT2-FLS-1 stands out as it has the best overall ranking. In Table 4.2, we compared the RMSE performance of the Z-GT2-FLS-1 with various other models [42–45] whose primary objective is to enhance accuracy performance. The comparison includes HTSK (i.e. a T1-FLS with more rules) [42], a Bayesian DL model [44], as well as other ML models. We can conclude that Z-GT2-FLS-1 achieves comparable RMSE measures (except for PP) while also being capable of generating HQ-PIs as presented in Table 4.2.

Table 4.2 : Testing RMSE: Z-GT2-FLS vs. Various Models.

Dataset	**Z-GT2-FLS-1**	**HTSK**[42]	**XGBoost**[42]	**MLP**[42]
PP	24.59(±0.73)	22.30(±0.19)	**18.93(±0.62)**	23.15(±0.15)
ABA	66.28(±1.83)	66.46(±0.85)	67.18(±0.85)	**63.59(±0.83)**
PM	**60.42(±4.13)**	82.89(±0.81)	85.61(±0.61)	78.56(±1.83)

Table 4.3 : Testing Performance Comparison of Dual-Focused FLSs over 20 Experiments for P = 5 rules.

Dataset	**Metric**	**Z-GT2-FLS-1**	**MJ-GT2-FLS-1 [30]**	**Z-GT2-FLS-2**	**MJ-GT2-FLS-2[30]**	**IT2-FLS-H[29]**	**IT2-FLS-HS[29]**
WW (11×4898)	#LP	192	247	192	247	225	280
	RMSE	**79.79(±1.45)**	81.44(±2.05)	80.48(±1.97)	82.02(±2.05)	82.39(±2.91)	82.62(±2.46)
	PICP	97.72(±0.62)	97.32(±1.14)	**97.82(±0.43)**	97.03(±1.49)	97.08(±1.52)	97.54(±1.58)
	PINAW	76.38(±8.07)	75.53(±10.7)	76.72(±7.60)	74.84(±11.09)	74.39(±11.40)	**74.35(±9.12)**
PP (4×9568)	#LP	73	93	73	93	85	105
	RMSE	24.59(±0.73)	24.56(0.73)	24.62(±0.74)	24.57(±0.74)	24.58(±0.69)	**24.37(±0.70)**
	PICP	98.74(±0.27)	98.73(±0.23)	98.74(±0.30)	98.72(±0.25)	98.71(±0.23)	**98.75(±0.20)**
	PINAW	30.88(±1.39)	30.50(±1.69)	30.92(±1.49)	30.46(±1.54)	**30.25(±1.33)**	30.80(±1.07)
ABA (8×4177)	#LP	141	181	141	181	165	205
	RMSE	66.28(±1.83)	67.01(±4.89)	67.07(±3.11)	**65.75(±1.61)**	67.89(±4.97)	66.40(±2.44)
	PICP	98.52(±0.71)	98.41(±0.44)	**98.58(±0.67)**	98.29(±0.43)	98.32(±0.39)	98.49(±0.38)
	PINAW	46.57(±4.18)	46.03(±4.27)	46.65(±4.29)	**45.99(±4.34)**	52.61(±4.30)	53.83(±4.40)
PM (19×5875)	#LP	328	423	328	423	385	480
	RMSE	60.42(±4.13)	75.23(±9.98)	**59.58(±5.01)**	75.45(±10.21)	88.14(±9.80)	74.17(±10.06)
	PICP	**98.95(±0.66)**	96.82(±3.81)	98.02(±0.71)	96.79(±3.28)	91.59(±4.30)	98.20(±2.88)
	PINAW	142.53(±29.25)	163.51(±34.05)	**103.92(±10.94)**	150.46(±28.05)	165.01(±36.49)	171.68(±39.58)

Table 4.3 (continued) : Testing Performance Comparison of Dual-Focused FLSs over 20 Experiments for P = 5 rules.

Dataset	Metric	Z-GT2-FLS-1	MJ-GT2-FLS-1 [30]	Z-GT2-FLS-2	MJ-GT2-FLS-2[30]	IT2-FLS-H[29]	IT2-FLS-HS[29]
AIDS	#LP	396	511	396	511	465	580
(23 × 2139)	RMSE	**69.73(±2.56)**	70.24(±2.91)	70.11(±2.34)	72.29(±3.26)	72.29(±2.75)	71.93(±3.52)
	PICP	97.52(±0.84)	94.85(±2.88)	96.84(±0.94)	95.29(±2.97)	97.00(±1.23)	**98.64(±0.69)**
	PINAW	160.35(±18.49)	160.69(±21.43)	**156.01(±20.60)**	164.77(±13.52)	185.57(±14.15)	210.75(±19.73)
Average	RMSE	**60.16**	63.70	60.37	64.02	67.16	63.90
	PICP	98.29	97.23	98.00	97.22	96.54	**98.32**
	PINAW	91.34	95.25	**82.84**	93.30	101.57	108.28
Average Rank	RMSE	**2.2**	3.2	3.2	3.6	5.2	3.4
	PICP	**1.8**	4.4	2.4	5.4	5.0	2.0
	PINAW	3.4	3.2	3.6	**2.6**	3.6	4.6
Overall Rank		**2.5**	3.6	3.1	3.9	4.6	3.3

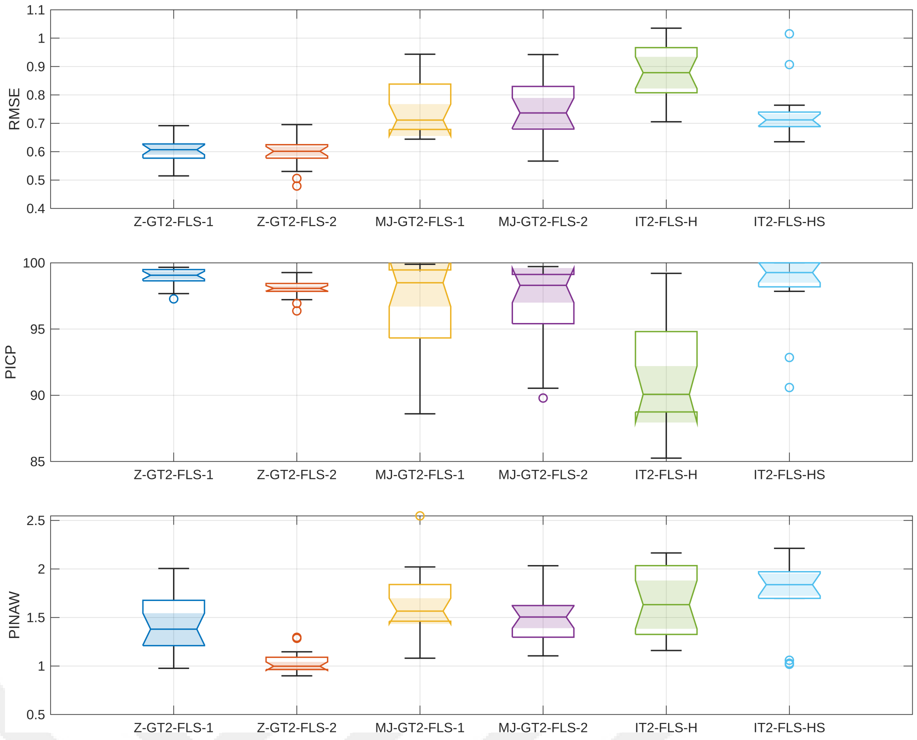


Figure 4.1 : Notched box-and-whisker plots for PM (19×5875) for $P = 5$.

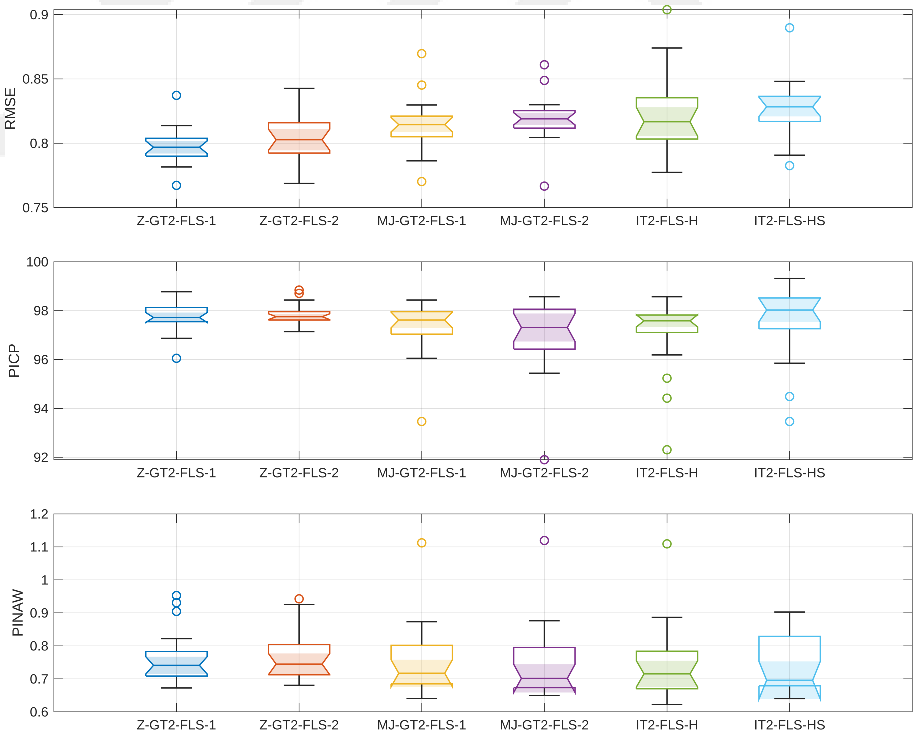


Figure 4.2 : Notched box-and-whisker plots for WW (11×4898) for $P = 5$.

4.2.2.2 Performance analysis for $P = 10$

In this subsection, we evaluate the performance of T2-FLSs for $P = 10$. In Figures 4.6, 4.7, 4.8, 4.9 and 4.10, for statistical analysis, we present the notched box and whisker

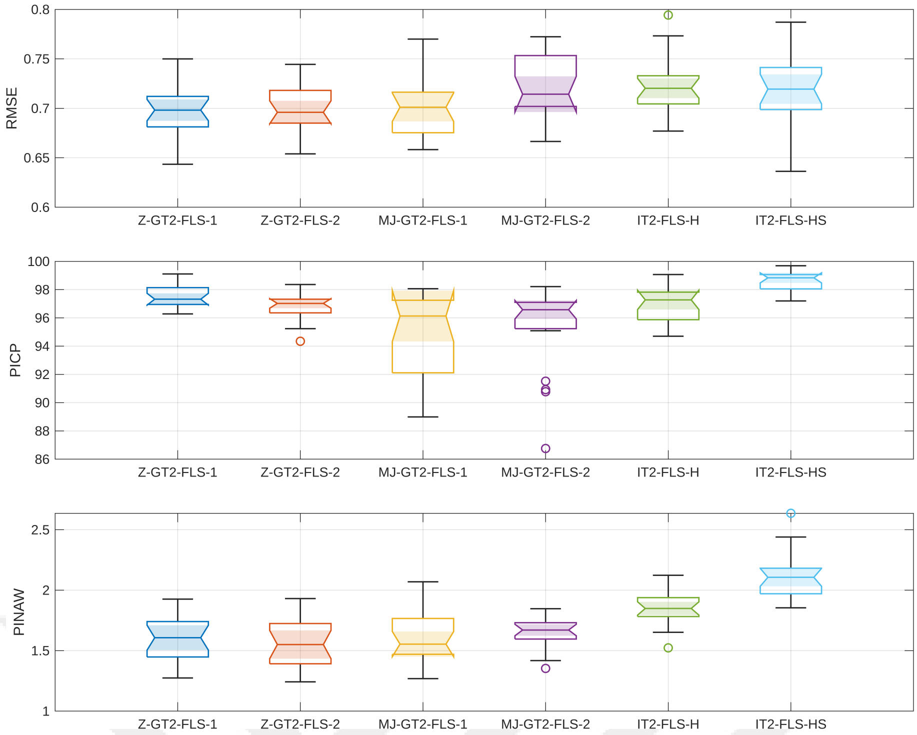


Figure 4.3 : Notched box-and-whisker plots for AIDS (23×2139) for $P = 5$.

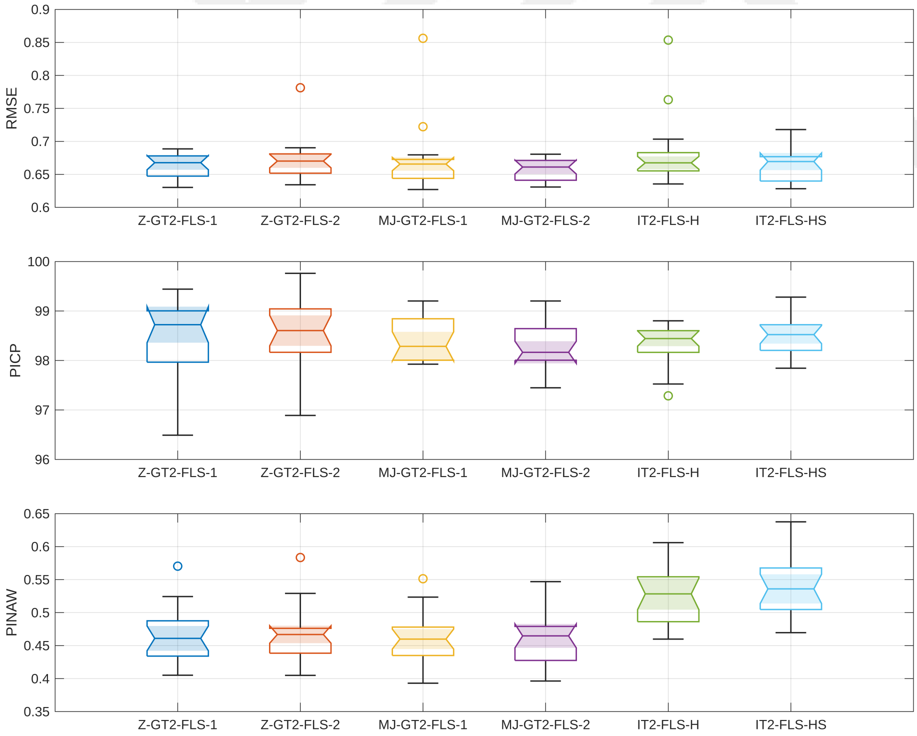


Figure 4.4 : Notched box-and-whisker plots for ABA (8×4177) for $P = 5$.

plots showing median (central mark), 25th/75th percentiles (left and right edges of box) which defines the IQR, whiskers (line), and outliers (circles). Observe that:

- For WW, the Z-GT2-FLS-1 outperforms the others in terms of both RMSE and PICP in Table 4.4. IT2-FLS-H stands out with the lowest PINAW value. We can conclude

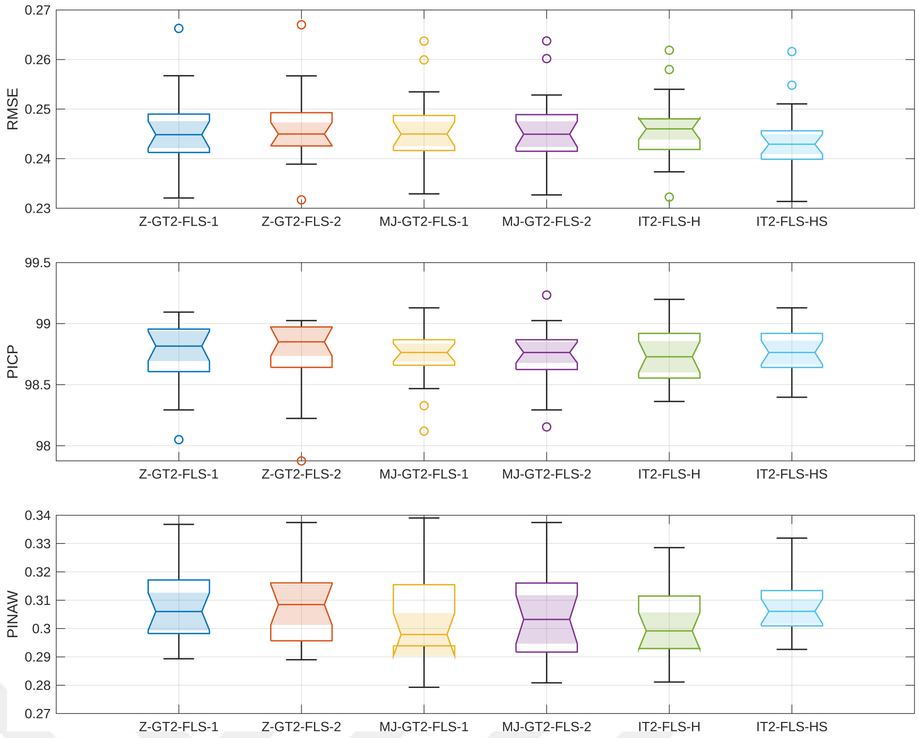


Figure 4.5 : Notched box-and-whisker plots for PP (4×9568) for $P = 5$.

that increasing P helps Z-GT2-FLS-1 to have the best results for both RMSE and PICP comparing Table 4.3. Also, while PICP increases, PINAW decreases, meaning HQ-PI[3].

- For PP, IT2-FLS-HS resulted in the best RMSE performance with a significant difference due to the absence of notches overlapping as shown in Figure 4.10. IT2-FLS-HS excels slightly in PICP, while MJ-GT2-FLS-1 has the lowest PINAW value in Table 4.4. Overall, Increasing P results in lower RMSE and higher PICP values compared to Table 4.3, and IT2-FLS-HS shows the best performance in terms of RMSE and PICP.

- For ABA, IT2-FLS-HS stands out as the best one in terms of RMSE and PICP, closely followed by other models in Table 4.4, while MJ-GT2-FLS-1 has the lowest PINAW. Generally, as shown in Figure 4.9, all models show similar performance for RMSE and PICP. GT2-FLSs have lower PINAW than IT2-FLSs and similar PICP values (i.e. HQ-PI). Overall, IT2-FLS-HS with $P = 10$ rules decreases RMSE and slightly increases PICP compared to IT2-FLS-HS with $P = 5$ rules in Table 4.3.

- For PM, Z-GT2-FLS-2 demonstrates superior performance with the lowest RMSE which is statistically significant when compared to MJ-GT2 and IT2 counterparts. Z-GT2-FLS models have similar performances in terms of RMSE and PICP as shown in

Figure 4.6. IT2-FLS-HS has the best coverage, yet its PINAW is significantly larger than other FLSs, indicating a poor PI. Z-GT2-FLS-2 with $P = 10$ rules decreases RMSE and PINAW compared to Z-GT2-FLS-2 with $P = 5$ rules in Table 4.3.

- For AIDS, IT2-FLS-HS has the best RMSE and PICP with the largest PI bands in Table 4.4. This results in a poor PI. Z-GT2-FLS-2 has the lowest PINAW. Furthermore, Z-GT2-FLS-1 with $P = 5$ rules, is the best performer in terms of RMSE across all FLSs in Table 4.3.

To sum up, Z-GT2-FLS-2, with minimal #LPs, is the best performer in average RMSE and PINAW, while Z-GT2-FLS-1 excels slightly in PICP, closely followed by IT2-FLS-HS. IT2-FLS-HS is the best performer in terms of average RMSE and PICP rank, yet its average PINAW rank is the worst, meaning poor PI. Z-GT2-FLS-2 and MJ-GT2-FLS-2 are the best performers for the average PINAW rank. Overall, Z-GT2-FLSs demonstrate competitive performance across various datasets by yielding accurate point-wise prediction while generating HQ-PI.

Table 4.4 : Testing Performance Comparison of Dual-Focused FLSs over 20 Experiments for P = 10 rules.

Dataset	**Metric**	**Z-GT2-FLS-1**	**MJ-GT2-FLS-1 [30]**	**Z-GT2-FLS-2**	**MJ-GT2-FLS-2[30]**	**IT2-FLS-H[29]**	**IT2-FLS-HS[29]**
WW (11×4898)	#LP	362	472	362	472	450	560
	RMSE	**79.52(±1.48)**	80.67(±1.78)	80.09(±1.88)	80.79(±2.02)	82.72(±3.69)	82.05(±1.78)
	PICP	**98.18(±0.44)**	97.81(±0.67)	97.88(±0.47)	97.82(±0.56)	96.91(±0.93)	97.51(±0.86)
	PINAW	72.38(±6.32)	72.44(±8.20)	71.88(±6.84)	72.00(±7.46)	**66.07(±4.59)**	68.44(±4.93)
PP (4×9568)	#LP	138	178	138	178	170	210
	RMSE	24.16(±0.72)	24.54(0.69)	24.19(±0.67)	24.53(±0.70)	23.66(±0.57)	**23.48(±0.57)**
	PICP	98.75(±0.27)	98.68(±0.24)	98.75(±0.29)	98.69(±0.28)	98.81(±0.31)	**98.84(±0.23)**
	PINAW	30.22(±1.45)	**29.89(±1.26)**	30.29(±1.51)	30.05(±1.23)	30.56(±1.39)	31.69(±1.24)
ABA (8×4177)	#LP	266	346	266	346	330	410
	RMSE	67.60(±4.58)	67.78(±8.25)	66.59(±1.91)	67.20(±4.03)	65.72(±1.39)	**65.31(±1.68)**
	PICP	98.57(±0.45)	98.27(±0.53)	98.59(±0.55)	98.42(±0.34)	98.52(±0.51)	**98.63(±0.40)**
	PINAW	46.35(±3.91)	**44.14(±3.41)**	45.41(±3.75)	44.39(±3.40)	51.57(±4.92)	52.72(±5.40)
PM (19×5875)	#LP	618	808	618	808	770	960
	RMSE	53.07(±5.25)	65.93(±5.28)	**50.02(±4.61)**	65.05(±4.74)	69.40(±14.65)	65.28(±4.52)
	PICP	99.41(±0.60)	98.80(±1.02)	98.30(±0.67)	98.73(±0.64)	97.06(±2.76)	**99.43(±0.67)**
	PINAW	128.20(±35.51)	144.51(±22.56)	**94.46(±11.65)**	123.01(±12.86)	145.57(±33.38)	173.39(±29.11)

Table 4.4 (continued) : Testing Performance Comparison of Dual-Focused FLSs over 20 Experiments for P = 10 rules.

Dataset	Metric	Z-GT2-FLS-1	MJ-GT2-FLS-1 [30]	Z-GT2-FLS-2	MJ-GT2-FLS-2[30]	IT2-FLS-H[29]	IT2-FLS-HS[29]
AIDS (23×2139)	#LP	746	976	746	976	930	1160
	RMSE	71.38(±2.71)	74.44(±3.16)	74.33(±3.14)	74.55(±3.39)	72.58(±3.25)	**69.82(±3.36)**
	PICP	98.01(±0.65)	96.42(±1.10)	97.41(±1.10)	96.71(±1.05)	97.15(±1.09)	**98.41(±0.83)**
	PINAW	163.36(±10.71)	163.59(±23.00)	**135.65(±10.64)**	161.35(±20.38)	199.12(±15.63)	234.99(±21.71)
Average	RMSE	59.15	62.67	**59.04**	62.42	62.82	61.19
	PICP	**98.58**	98.00	98.19	98.07	97.69	98.56
	PINAW	88.10	90.91	**75.54**	86.16	98.58	112.25
Average Rank	RMSE	2.6	5.0	2.8	4.4	3.8	**2.4**
	PICP	2.2	5.0	3.2	4.4	4.4	**1.8**
	PINAW	3.6	3.2	**2.4**	**2.4**	4.2	5.2
Overall Rank		**2.8**	4.4	**2.8**	3.7	4.3	3.1

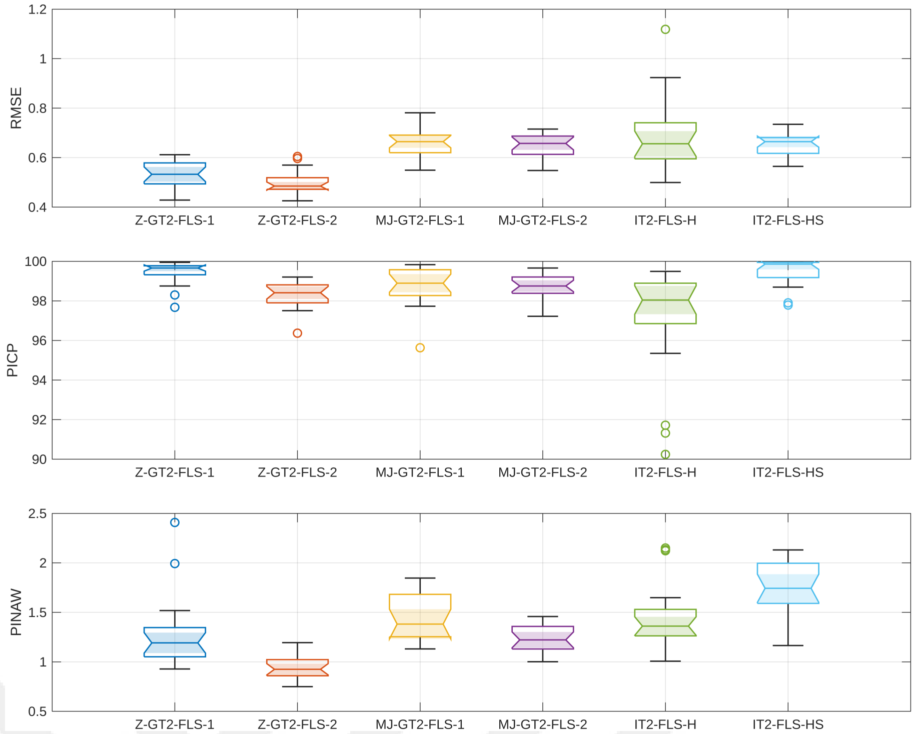


Figure 4.6 : Notched box-and-whisker plots for PM (19 × 5875) for $P = 10$.

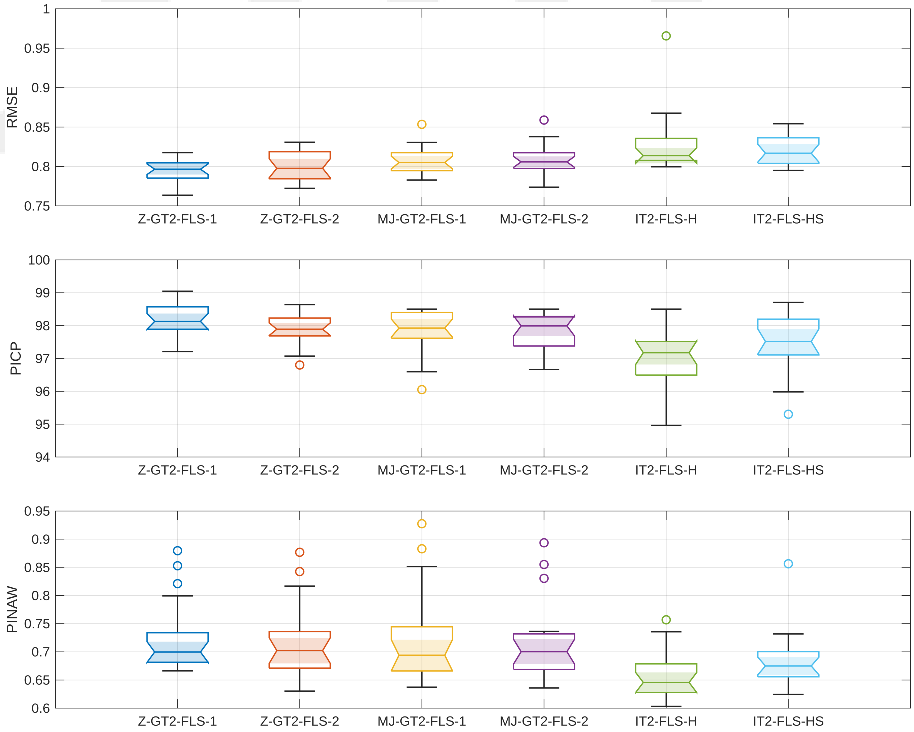


Figure 4.7 : Notched box-and-whisker plots for WW (11 × 4898) for $P = 10$.

4.2.2.3 Computational load analysis

Here, we analyze the computational load of training the models with $P = 5$ rules and $P = 10$ rules, respectively. To analyze the training complexity, the computational load

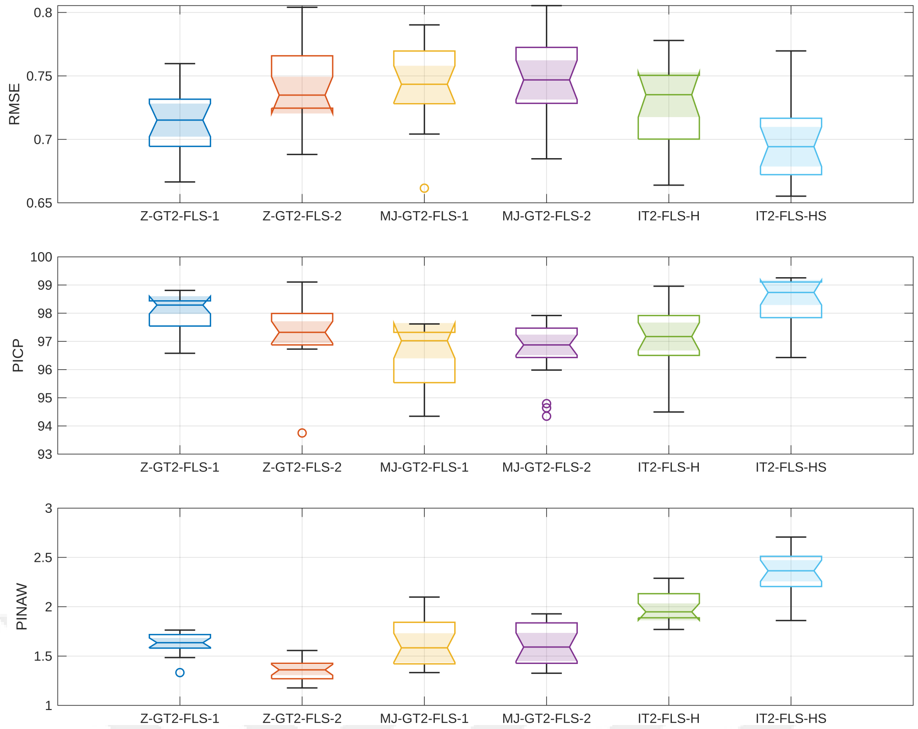


Figure 4.8 : Notched box-and-whisker plots for AIDS (23 × 2139) for $P = 10$.

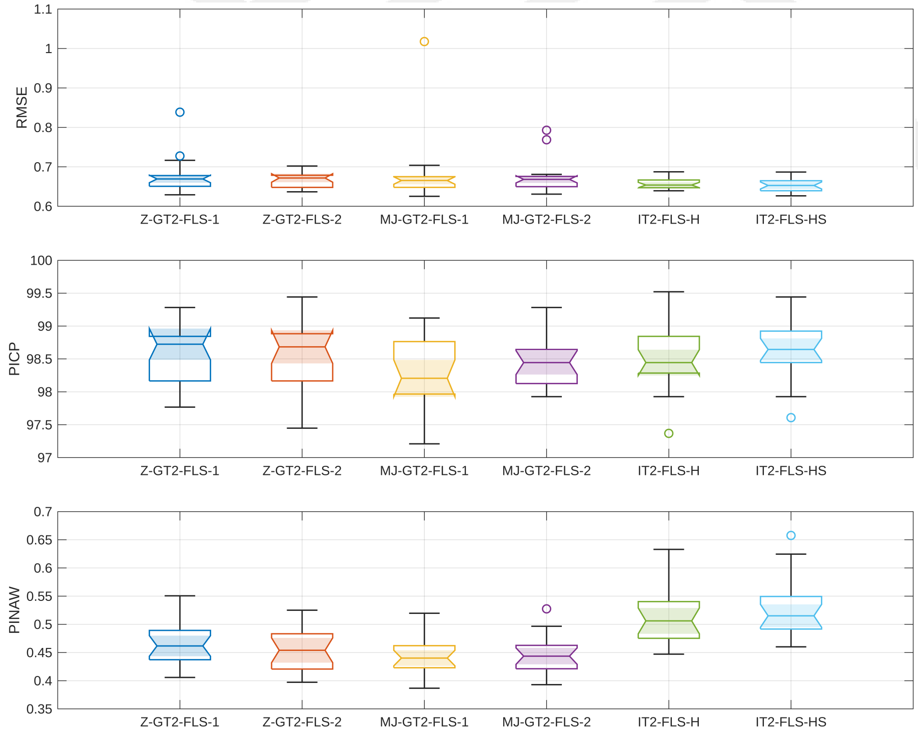


Figure 4.9 : Notched box-and-whisker plots for ABA (8 × 4177) for $P = 10$.

during training is examined by recording the Memory Usage (MU) and total Training Time (TT). Table 4.5 provides the MU and Training Time TT associated with each model composed with $P = 5$ and $P = 10$ rules. We also provided the rankings over the handled 5 datasets for an easy comparison. These results were obtained using a

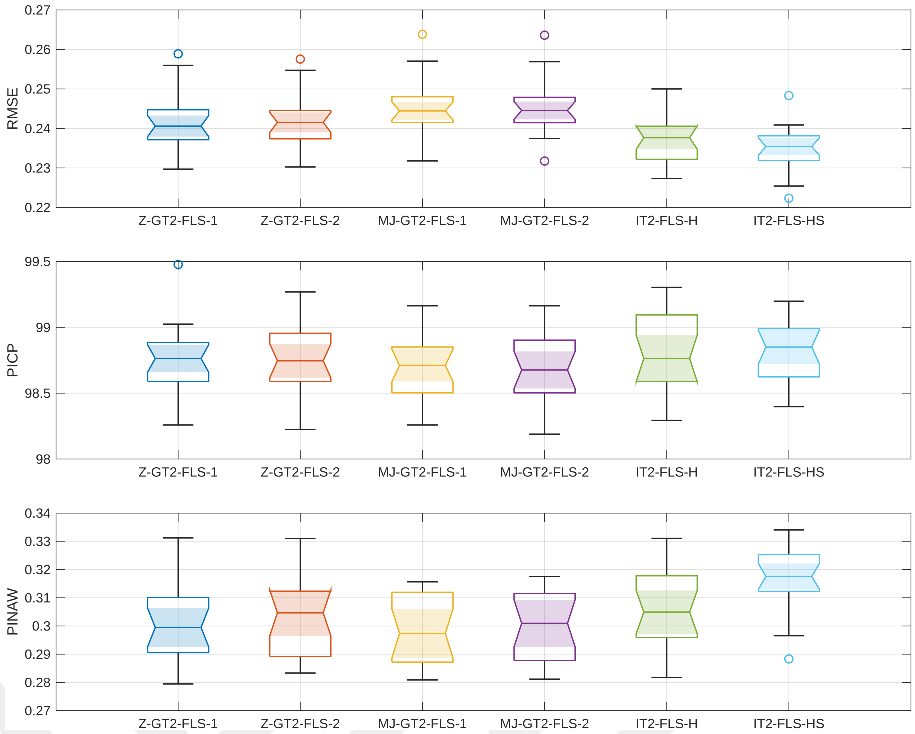


Figure 4.10 : Notched box-and-whisker plots for PP (4×9568) for $P = 10$.

computer equipped with a NVIDIA GTX 1080 TI GPU.

We can observe from Tables 4.5 and 4.6 that as expected increasing the rule size from 5 to 10 has significantly increased the MU and TT values for both the IT2 and GT2 FLSs. Especially, the computational loading (MU) and TT of the GT2-FLSs have increased since they are constructed by a collection of α-planes (α_k), and the output of GT2-FLSs is defined with a weighted average of outputs of each α_k-IT2-FLS via (2.25). Increasing the rule number to $P = 10$ significantly raises MU across all datasets, but less so for TT. Z-GT2-FLSs have similar MU with MJ-GT2-FLSs, yet their TTs are generally less than MJ-GT2 counterparts due to the fewer #LPs for both $P = 5$ and $P = 10$. Additionally, the memory usage of IT2-FLS-H is similar to that of IT2-FLS-HS, yet IT2-FLS-H consistently requires less training time than IT2-FLS-HS across all datasets.

Table 4.5 : Computational Load Analysis over 20 Experiments for *P* = 5 Rules.

Dataset	**Metric**	**Z-GT2-FLS-1**	**MJ-GT2-FLS-1**	**Z-GT2-FLS-2**	**MJ-GT2-FLS-2**	**IT2-FLS-H**	**IT2-FLS-HS**
WW	MU (MB)	197	197	198	196	**182**	**182**
	TT (s)	47	48	46	48	**39**	42
PP	MU (MB)	209	208	210	208	**186**	**186**
	TT (s)	25	26	25	25	**19**	23
ABA	MU (MB)	188	189	190	188	**180**	**180**
	TT (s)	78	81	78	81	**63**	71
PM	MU (MB)	222	220	222	220	**192**	193
	TT (s)	161	167	160	166	**131**	148
AIDS	MU (MB)	190	188	190	190	**180**	**180**
	TT (s)	39	41	38	40	**30**	34
Average	MU (MB)	201.2	200.4	202	200.4	**184**	184.2
	TT (s)	70	72.6	69.4	72	**56.4**	63.6
Average Rank	MU (MB)	3	2.6	3.8	2.4	**1**	1.2
	TT (s)	3.6	5.0	3	4.4	**1**	2
Overall Rank		3.3	3.8	3.4	3.4	**1**	1.6

Table 4.6 : Computational Load Analysis over 20 Experiments for P = 10 Rules.

Dataset	Metric	Z-GT2-FLS-1	MJ-GT2-FLS-1	Z-GT2-FLS-2	MJ-GT2-FLS-2	IT2-FLS-H	IT2-FLS-HS
WW	MU (MB)	402	400	402	400	**254**	**254**
	TT (s)	54	55	53	56	**40**	43
PP	MU (MB)	611	610	612	610	**328**	**328**
	TT (s)	30	31	30	30	**20**	25
ABA	MU (MB)	353	352	352	352	**234**	**234**
	TT (s)	90	94	91	92	**65**	72
PM	MU (MB)	538	534	538	534	**302**	306
	TT (s)	197	205	194	198	**136**	151
AIDS	MU (MB)	289	288	288	288	**222**	**222**
	TT (s)	46	47	45	46	**32**	36
Average	MU (MB)	438.6	436.8	438.4	436.8	**268**	268.8
	TT (s)	83.4	86.4	82.6	84.4	**58.6**	65.4
Average Rank	MU (MB)	3.2	2.2	3	2.2	**1**	1.2
	TT (s)	3.6	5.2	3.2	4.6	**1**	2
Overall Rank		3.4	3.7	3.1	3.4	**1**	1.6

5. LEARNING GT2-FLSs FOR DISTRIBUTION ESTIMATION

While the dual-focused Z-GT2-FLS learning method offers improved predictive performance for a specific confidence level, it requires retraining for each target quantile pair $[\underline{\tau}, \overline{\tau}]$ for a given desired confidence level.

In this chapter, we introduce a DL framework to train GT2-FLSs to learn the conditioned inverse cumulative function of a given dataset $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, by estimating all quantile levels $\tau \in [0,1]$. To achieve this, we utilize the SQR approach [12] that aims to estimate all quantile levels simultaneously with:

$$\min_{\boldsymbol{\theta} \in C} L = \frac{1}{N} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\tau \sim U[0,1]} \left[\rho \left(y\left(\boldsymbol{x}_n\right), y_n, \tau \right) \right] \tag{5.1}$$

The predictive distribution is learned by randomly sampling quantile levels τ from a uniform distribution $\tau \sim U[0,1]$ for each data point and mini-batch during training [12]. Within this framework, we first introduce the integration of the SQR method into GT2-FLSs. We then present Adaptive SQR (ASQR), an enhanced version of SQR that targets miscalibrated regions by focusing sampling in those areas during training, thereby improving the generalization of the inverse cumulative distribution of a given dataset.

5.1 SQR For GT2-FLS

Here, rather than only using α_0-IT2-FLS for UQ, we designate an α-IT2-FLS to represent a quantile level τ. We start by eliminating the output calculation of GT2-FLS defined in (2.25) and define the output as $y^{\alpha}(\boldsymbol{x}), \alpha \in [0,1]$. Then, we transform $y^{\alpha}(\boldsymbol{x})$ as:

$$y^{\alpha}(\boldsymbol{x}) \rightarrow y(\boldsymbol{x}, \alpha), \forall \alpha \in [0,1] \tag{5.2}$$

Thus, $\alpha \in [0,1]$ has been transformed from a structural parameter to an input argument of GT2-FLS that defines which α-IT2-FLS is used as the output of the GT2-FLS.

To enforce that α-IT2-FLS learns a quantile level τ, we match α-planes with the quantile levels selected randomly from the uniform distribution $\tau \sim U[0,1]$ of SQR, i.e. $\alpha = \tau$.

The defined learning problem for GT2-FLS is then as follows:

$$\min_{\boldsymbol{\theta} \in C} L = \frac{1}{N} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\tau \sim U[0,1]} \left[\rho \left(y \left(x_n, \tau \right), y_n, \tau \right) \right] \tag{5.3}$$

Thus, after training, α-IT2-FLS represents a quantile level function, providing flexibility to generate any desired quantile level $\tau \in [0, 1]$ by setting $\alpha \in [0, 1]$ as $\alpha = \tau$ during inference.

While training, as $\tau \sim U[0, 1]$, we also deploy a validation phase defined with the Expected Calibration Error (ECE) [11]:

$$ECE(\tau, \hat{\tau}) = \frac{1}{N} \sum_{i=1}^{N} E_i \tag{5.4}$$

where E_i is the calibration error defined as:

$$E_i = |\tau_i - \hat{\tau}_i| \tag{5.5}$$

Here, $\hat{\tau}$ defines the estimated quantile levels resulting from the GT2-FLS while τ are discretized from 0.01 to 0.99 with a 0.01 increment. Training is stopped early if the ECE does not decrease for more than T_{st} epochs, until a maximum of T epochs. If training is stopped early, the final GT2-FLS is backtracked to the one with the lowest ECE value [11].

5.2 ASQR For GT2-FLSs

The efficacy of the SQR-based learning method for GT2-FLS is intricately tied to the stochastic selection of τ for individual samples and mini-batches during training. While achieving satisfactory distribution estimation is possible by increasing the mini-batch size, this comes at the cost of increased computational burden. In this context, we present an Adaptive SQR (ASQR) with a focus on mitigating miscalibration. The algorithm in Table 5.1 provides the ASQR training steps for GT2-FLSs.

The proposed adaptation enhancement for SQR runs in the validation phase and orients the randomly generated τ's to spaces, resulting in large calibration errors E (defined in (5.5)). As given in the algorithm in Table 5.2, we first calculate E for each quantile level and define a deficient quantile estimation as $C : E \geq \epsilon$, where ϵ is a threshold. We then obtain the miscalibration areas $A = (A_1, \ldots, A_N)$, and $[\underline{\tau}_A, \overline{\tau}_A]$ to be used for producing quantile samples. These miscalibration areas are based on the As presented

Table 5.1 : ASQR to learn Z-GT2-FLS Algorithm.

Step	Description
1	**Input:** N training samples $(x_n, y_n)_{n=1}^{N}$, $\phi = [\underline{\tau}, \overline{\tau}]$
2	Set P: number of rules
3	Set mbs: mini-batch size
4	Set T: number of epochs
5	Set ϵ : threshold value
6	**Output:** Learned parameter set θ_{GT2}
7	Initialize $\theta_{GT2} = [\theta_{GT2-A}, \theta_{GT2-C}]$
8	$\alpha_k \sim U[0,1]$
9	**For** $t = 1$ to T **do**
10	**For each** mbs in N **do**
11	Perform parametrization tricks for θ (see Sec. 3.3.3)
12	$\mu^* \leftarrow \mathrm{PMF}(x; \theta_{AP})$ (3.11)
13	$[\underline{\mu}^{\alpha_k}, \overline{\mu}^{\alpha_k}] \leftarrow \mathrm{SMF}(\mu^*; \theta_{AS})$ (3.15) and (3.16)
14	$y^{\overline{\alpha_k}} \leftarrow \mathrm{Inference}(\underline{\mu}^{\alpha_k}, \overline{\mu}^{\alpha_k}; \theta_C)$ (see Sec. 2.2)
15	Compute L (5.3)
15	Compute $\partial L / \partial \theta$ via automatic differentiation (AD)
16	Update θ via a DL optimizer, e.g., Adam
17	**End for**
18	Compute $\hat{\tau}$ and ECE on Validation Dataset
19	$[A, \underline{\tau}_A, \overline{\tau}_A] \leftarrow \mathrm{MiscalibrationSpace}(\tau, \hat{\tau}; \epsilon)$
20	$\alpha_k \leftarrow \mathrm{QuantileGeneration}(\tau, \hat{\tau}, \underline{\tau}_A, \overline{\tau}_A, A)$
21	**End for**
22	$\theta^* \leftarrow \arg\min L$
23	**Return:** θ^*

in the Algorithm in Table 5.3, for each A_i, the normalized error $\tilde{e}_{A_i}$ is computed, and subsequently, quantile samples are curated based on the magnitude of miscalibration.

In conclusion, the ASQR produces more samples from the regions where the miscalibration area is more substantial. This process is integral to enhancing the learning performance of GT2-FLS for predictive distribution estimation, as more focus is provided to estimation errors during training.

5.3 Comparative Performance Analysis

We show the learning performance of the Z-GT2-FLS compared to state-of-the-art DL methods for UQ. To facilitate a direct comparison with the results of [11], we considered the PP, Red Wine (RW), Concrete Strength (CS), Boston Housing (BH), Naval Propulsion Plant (NP), Kin8nm, Yacht Hydrodynamics (YH), and Energy Efficiency (EE) datasets. For the completeness of the study, we also handled ABA,

Table 5.2 : Miscalibration Space Algorithm.

Step	Description
1	**Input:** Pairs $(\tau_i, \hat{\tau}_i)_{i=1}^{N}$, threshold ϵ
2	Initialize empty sets $A \leftarrow \emptyset$, $A_k \leftarrow \emptyset$
3	Initialize empty interval lists $\underline{\tau}_A \leftarrow \emptyset$, $\overline{\tau}_A \leftarrow \emptyset$
4	Compute errors $E = \{ e_i = \lvert \tau_i - \hat{\tau}_i \rvert : i = 1, \ldots, N \}$
5	**For each** $e_i \in E$ **do**
6	Mask small errors: $e_i \leftarrow e_i \cdot \mathbb{I}\{e_i \geq \epsilon\}$
7	**If** $e_i \neq 0$ **then**
8	Add to current miscalibration group: $A_k \leftarrow A_k \cup \{(e_i, \tau_i)\}$
9	**Else**
10	Close group A_k: $A \leftarrow A \cup \{A_k\}$
11	Record interval: $\underline{\tau}_A \leftarrow \underline{\tau}_A \cup \{\min(\tau_{A_k})\}$, $\overline{\tau}_A \leftarrow \overline{\tau}_A \cup \{\max(\tau_{A_k})\}$
12	Reset group: $A_k \leftarrow \emptyset$
13	**End If**
14	**End For**
15	**Return:** A, $\underline{\tau}_A$, $\overline{\tau}_A$

Table 5.3 : Quantile Generation Algorithm.

Step	Description
1	**Input:** τ, $\hat{\tau}$, intervals $\underline{\tau}_A$, $\overline{\tau}_A$, groups A
2	Initialize empty set $\alpha_A \leftarrow \emptyset$
3	**For each** miscalibration group $A_i \in A$ **do**
4	Compute normalized weight: $\tilde{e}_{A_i} \leftarrow \dfrac{e_{A_i}}{\sum_{j=1}^{\lvert A \rvert} e_{A_j}}$
5	Determine sample count: $ns \leftarrow \lfloor \tilde{e}_{A_i} \times mbs \rfloor$
6	Sample quantile(s): $\alpha_{A_i} \sim U\left[\underline{\tau}_{A_i}, \overline{\tau}_{A_i}\right]$ (draw ns samples)
7	Aggregate: $\alpha_A \leftarrow \alpha_A \cup \{\alpha_{A_i}\}$
8	**End For**
9	**Return:** α_A

PM, and AIDS, which were utilized in Section 4.2.

5.3.1 Design of experiments

All input features are standardized via Z-score normalization:

$$\tilde{x}_i = \frac{x_i - \mu_x}{\sigma_x} \tag{5.6}$$

where μ_x and σ_x are the sample mean and standard deviation of feature x over the full dataset. The target variable y is similarly normalized on the *training set*:

$$\tilde{y} = \frac{y - \mu_y}{\sigma_y} \tag{5.7}$$

where μ_y and σ_y denote the sample mean and standard deviation of y. After inference, predictions are mapped back via

$$y = \tilde{y}\sigma_y + \mu_y \tag{5.8}$$

Each dataset is randomly split into a training set (90%) and a test set (10%), Within the training set, 20 % was used for validation as in [11]. All models (Z-GT2-FLS, MJ-GT2-FLS, and the two IT2-FLSs) are trained with identical hyperparameters:

- Mini-batch size: 64
- Learning rate: 1×10^{-2}
- Number of epochs: 10000
- $T_{st} = 200$

For each dataset, we trained a Z-GT2-FLS via SQR (Z-GT2-SQR) and one via ASQR (Z-GT2-ASQR) with $P = 10$ rules. If the validation ECE loss did not show improvement of more than $T_{st} = 200$ epochs, we terminated the training section early and used the model with the lowest ECE loss as stated in [5]. For the Z-GT2-ASQR, ϵ is selected by performing cross-validation with $\epsilon \in \{0.005, 0.01, 0.02, 0.03, 0.04, 0.05\}$. The experiments were conducted within MATLAB ® and repeated with 5 different initial seeds for statistical analysis.

5.3.2 Performance evaluation

To assess the estimations, we calculated the ECE measure defined in (5.4). The comparative outcomes on the test datasets, generated through 5 experiments, are presented in Tables 5.4 and 5.5. Note that all compared models are neural networks consisting of 2 layers with 64 hidden units each, utilizing ReLU activation function [11]. We have included rankings across the datasets for ease of comparison. We also presented the Quantile-Quantile (Q-Q) plots of Z-GT2 SQR and ASQR in Figures 5.1, 5.2, 5.3, and 5.4 which serve as a means of comparing expected quantile levels against observed ones.

Z-GT2-FLS, particularly Z-GT2-ASQR, exhibit competitive performance compared to state-of-the-art DL UQ methods. As shown in Table 5.4, their ability to calibrate predictions is evident across multiple datasets. Despite MAQR [11] being acknowledged as the top-performing DL method, Z-GT2-ASQR consistently outperforms MAQR across various datasets, as indicated by its average rank. This showcases the robustness of our novel approach. Thus, Z-GT2-ASQR is capable of achieving superior calibration irrespective of the data characteristics. It is noteworthy that the performance of Z-GT2-SQR is also commendable, as it surpasses four out of five DL methods. Also, Z-GT2-ASQR offers a more robust and consistent performance based on Table 5.5.

To examine the impact of the adaptation mechanism on distribution estimation, we presented the Q-Q plots from a single experiment obtained across datasets with diverse sizes and dimensions in Figures 5.1, 5.2, 5.3, and 5.4. The visual results underscore the notable performance improvement introduced by ASQR, with Z-GT2-ASQR exhibiting enhanced adherence to the diagonal in the Q-Q plots compared to its SQR counterpart. This improvement is a result of the quantile sampling strategy of Z-GT2-ASQR, which enforces training more inadequately learned quantile levels through a targeted sampling process. It is also worth underlining that the most striking improvement is observed in Figure 5.4, where ASQR significantly outperforms SQR, especially in small quantile level prediction.

Table 5.4 : ECE Comparison over 5 Experiments: Z-GT2-SQR and Z-GT2-ASQR vs. Various Models.

Dataset ($M \times N$)	SQR [12]	mPAIC [13]	Interval [11]	Cali [11]	MAQR [11]	Z-GT2-SQR	Z-GT2-ASQR
PP (4×9568)	2.6(±0.4)	5.2(±0.4)	2.2(±0.4)	2.0(±0.1)	**1.6(±0.3)**	1.9(±0.6)	1.6(±0.5)
RW (11×1599)	4.2(±0.2)	10.3(±0.3)	5.0(±0.8)	4.2(±0.4)	2.7(±0.3)	4.0(±1.5)	**2.3(±0.4)**
CS (8×1030)	9.3(±1.5)	6.2(±0.5)	**3.7(±0.6)**	5.6(±0.8)	5.3(±0.4)	4.3(±1.7)	4.2(±1.1)
BH (13×506)	9.0(±0.8)	8.7(±1.3)	6.9(±1.1)	8.5(±1.5)	6.2(±1.8)	5.4(±0.9)	**4.8(±0.9)**
NP (17×11937)	9.7(±1.6)	3.1(±0.5)	4.7(±1.4)	5.9(±0.7)	2.3(±0.2)	2.1(±0.4)	**1.9(±0.9)**
Kin8nm (8×8198)	4.4(±0.1)	6.6(±0.4)	2.9(±0.4)	3.5(±0.3)	1.8(±0.4)	2.1(±0.4)	**1.6(±0.4)**
YH (6×308)	9.4(±0.9)	10.8(±2.3)	7.5(±0.9)	8.3(±0.6)	6.8(±2.1)	**6.3(±2.5)**	8.6(±1.8)
EE (8×768)	9.8(±0.8)	10.4(±0.5)	4.3(±0.6)	5.8(±0.4)	**3.5(±1.0)**	6.2(±2.8)	4.2(±0.7)
Average	7.3	7.7	4.7	5.5	3.8	4.1	**3.7**
Average Rank	6.1	6.4	3.9	4.7	2.3	2.8	**1.9**

Table 5.5 : ECE Comparison over 5 Experiments: Z-GT2-SQR vs. Z-GT2-ASQR.

Dataset ($M \times N$)	Z-GT2-SQR	Z-GT2-ASQR
WW (11 × 4898)	1.7(±0.4)	**1.7(±0.3)**
ABA (8 × 4177)	1.7(±1.3)	**1.6(±0.4)**
PM (19 × 5875)	1.7(±0.7)	**1.7(±0.4)**
AIDS (23 × 2139)	**4.3(±0.8)**	4.8(±1.3)
Average	**2.3**	2.4
Average Rank	1.8	**1.3**

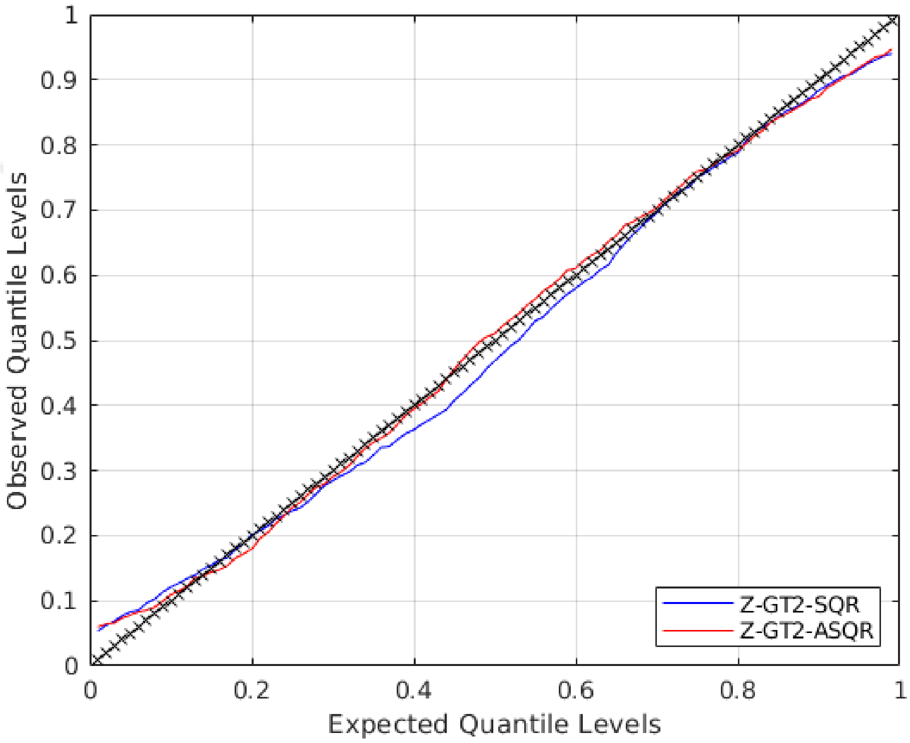


Figure 5.1 : Q-Q plot for Kin8nm (8 × 8198).

5.3.3 ASQR implementation - visualization

Here, we dive deep into how the ASQR algorithm works properly by showing an example on the BH dataset.

First, we implement the SQR algorithm [12], which selects quantile levels randomly in each epoch. We show the Q-Q plots of Z-GT2-SQR in Figure 5.5, which serve as a means of comparing expected quantile levels against observed ones.

As illustrated in Figure 5.5, the model struggles to accurately estimate the desired quantile levels. Specifically, the mean ECE for the BH dataset across 5 experiments is 5.4. This value will serve as a reference point for comparison with the proposed

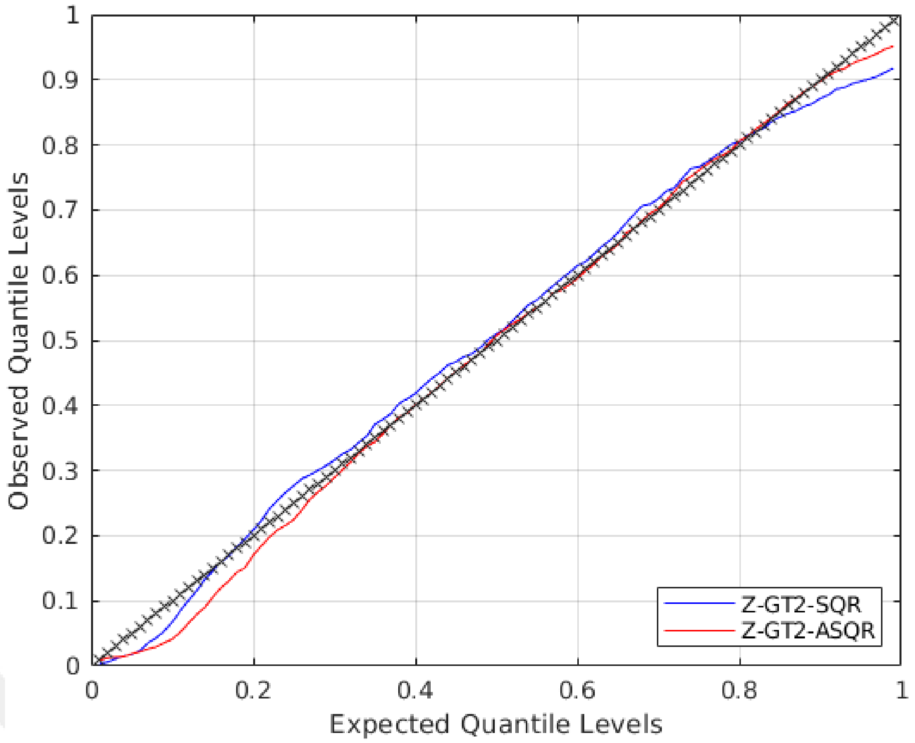


Figure 5.2 : Q-Q plot for Naval (17 × 11937).

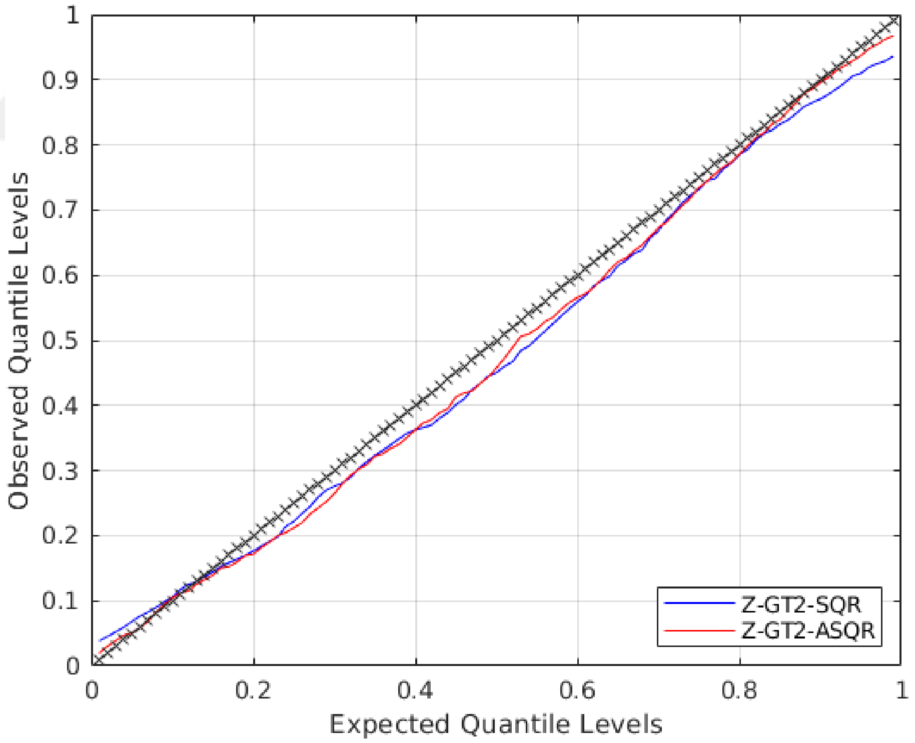


Figure 5.3 : Q-Q plot for Power (4 × 9568).

Z-GT2-ASQR method. To handle this problem, we propose the ASQR method as in Section 5.2. Now, we present the ASQR algorithm with visualizations on the BH dataset.

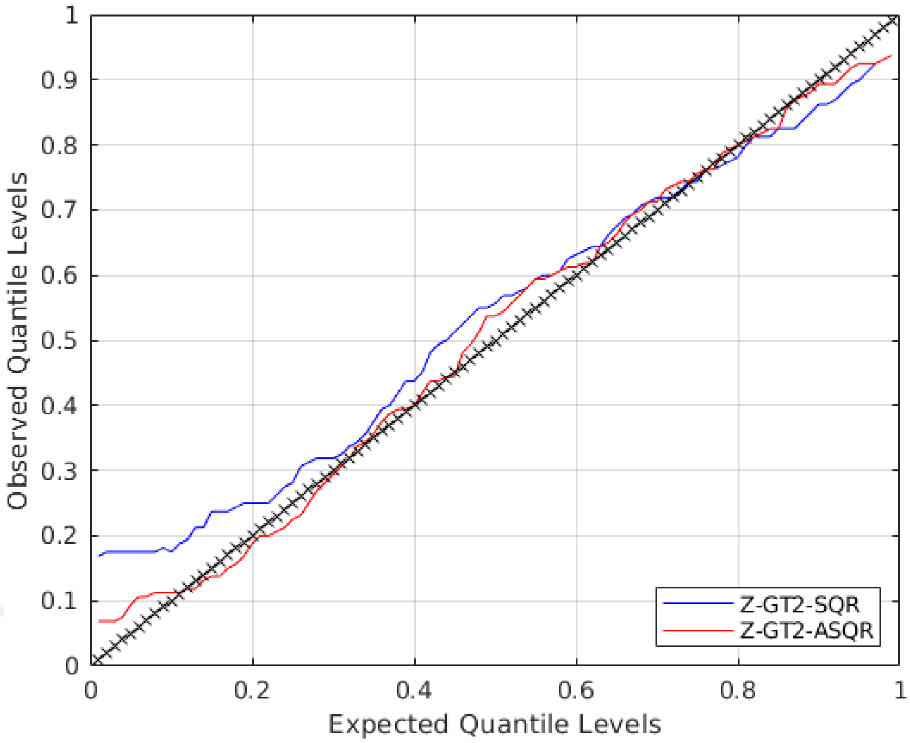


Figure 5.4 : Q-Q plot for RW (11 × 1599).

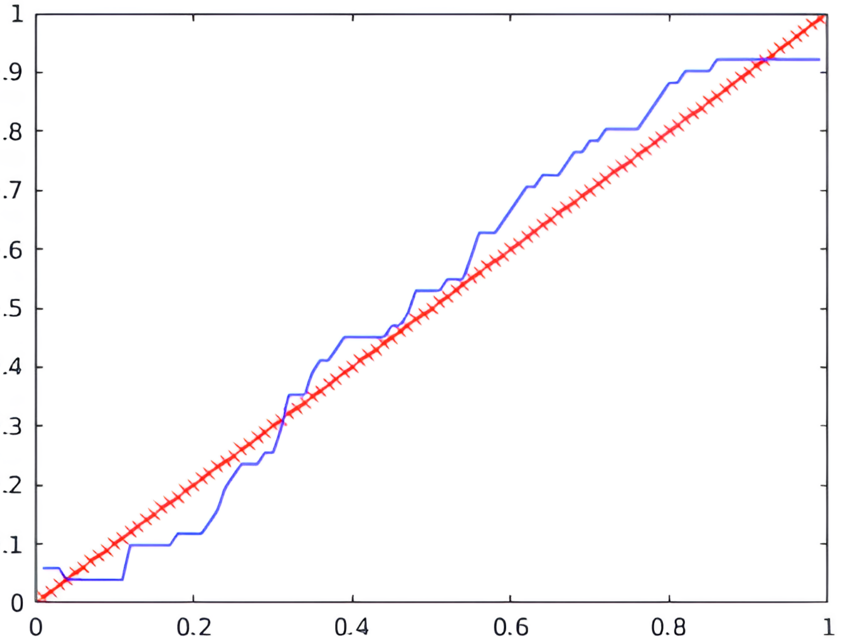


Figure 5.5 : Q-Q plot for BH with Z-GT2-SQR method for 1 seed.

- We calculate the absolute error (E) for all quantile levels $\tau = [0.01, \ldots, 0.99]$ during the validation step as:

$$E = |\tau - \hat{\tau}| \tag{5.9}$$

- We define a Condition (C) as:

$$C : E > \epsilon \tag{5.10}$$

  where ϵ is a threshold.

- According to C, we define areas, which are the above of the threshold value ϵ, and call them miscalibration areas, represented as $A_1, \ldots, A_n$ as shown in Figure 5.6.

- For each A_i, we find minimum and maximum quantile levels $(\underline{\tau}_{A_i}, \overline{\tau}_{A_i})$, then,

$$\tau_{A_i} = (\overline{\tau}_{A_i} - \underline{\tau}_{A_i})U[0,1] + \underline{\tau}_{A_i} \tag{5.11}$$

$$\tau_A = \cup \tau_{A_i} \tag{5.12}$$

- With this selection of τ:

$$\tau \sim (\overline{\tau}_A - \underline{\tau}_A)U[0,1] + \underline{\tau}_A \tag{5.13}$$

  Lastly we assign $\alpha = \tau$.

The density of selected τ levels depend on the miscalibration areas $A_1, \ldots A_n$ as represented in Table 5.3. Figure 5.7 shows the selected quantile levels ($\tau = \alpha$) from the miscalibration areas, as depicted in Figure 5.6.

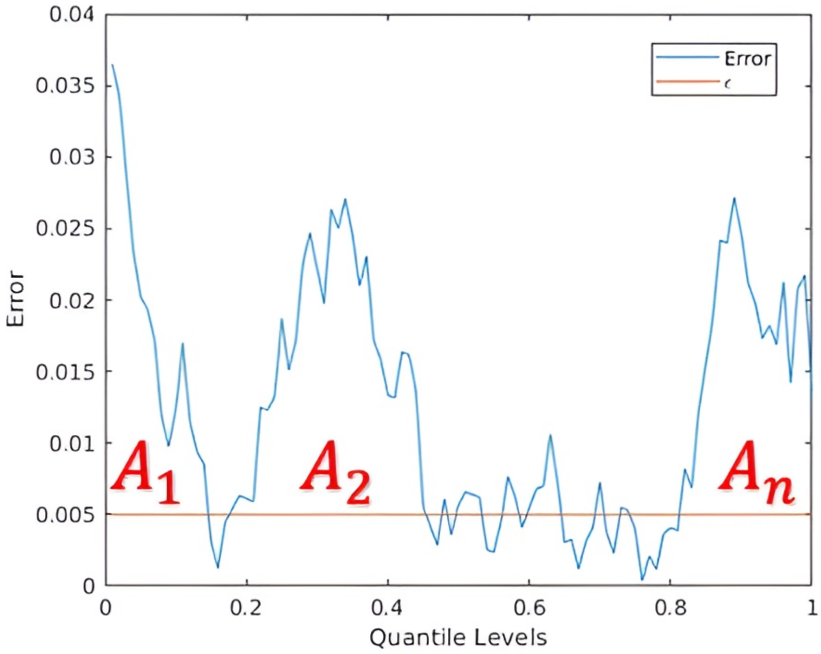


Figure 5.6 : Miscalibration Areas for $\epsilon = 0.005$.

Afterwards, we show the Z-GT2 ASQR Q-Q plot in Figure 5.8. The ASQR method achieves a mean ECE of 4.8 on the BH dataset across five experiments, representing an improvement of 0.6 compared to the SQR method. We can conclude that using the ASQR method helps us to select quantile levels, which are not well-calibrated during the validation step. This provides us with better generalization for estimating the inverse cumulative distribution.

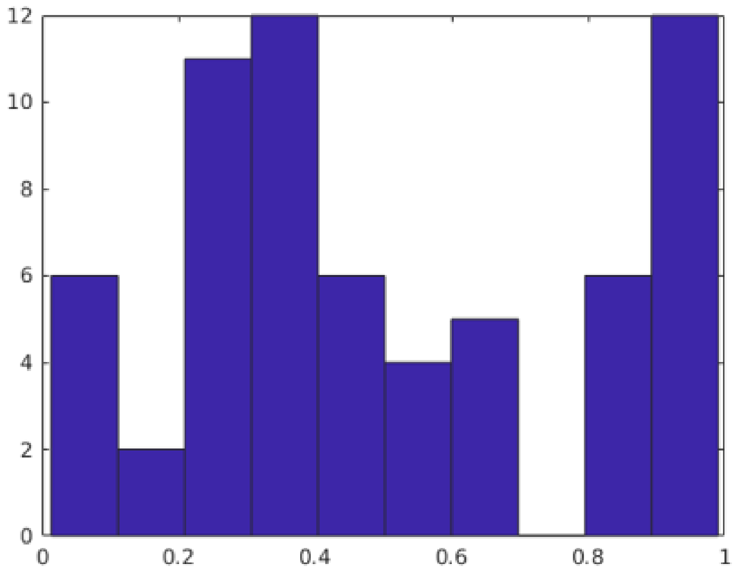


Figure 5.7 : Selected $\alpha = \tau$ levels.

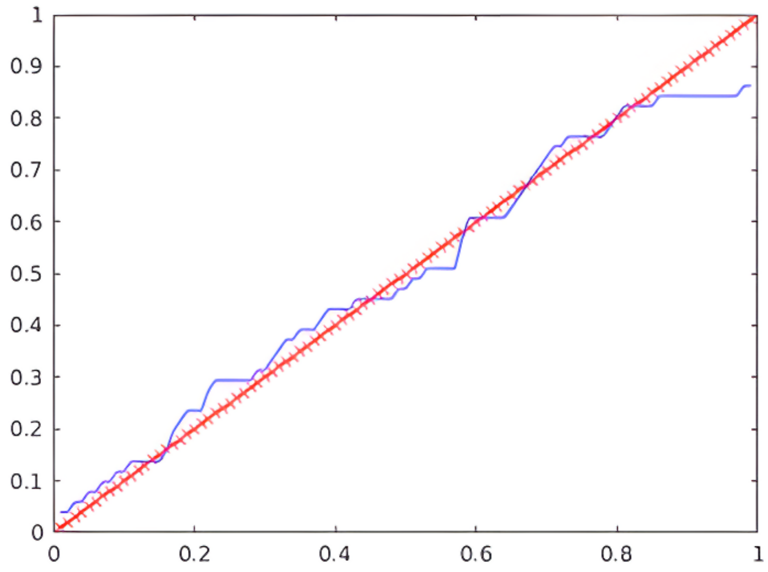


Figure 5.8 : Q-Q plot for BH with Z-GT2-ASQR method for 1 seed.

6. CONCLUSIONS AND RECOMMENDATIONS

In this master's thesis, we present a DL framework for Z-GT2-FLS, a novel approach grounded in Zadeh's original definition of GT2-FSs. This framework is designed to excel in delivering both highly accurate point-wise predictions and robust UQ. We introduce an approach to combine Z-GT2-FSs with α-plane representation to design Z-GT2-FLSs. Additionally, we propose a dual-focused DL framework that leverages Z-GT2-FLSs to simultaneously optimize for prediction accuracy and uncertainty modeling. Furthermore, we propose novel methods/algorithms for learning inverse cumulative distribution via Z-GT2-FLSs by estimating all quantile levels simultaneously.

In the initial chapter of this thesis, we provide the mathematical foundations of T2-FSs and T2-FLSs. We outline the core components of IT2/GT2-FLSs, which are rule structure, antecedent and consequent MFs, type reduction, and defuzzification, to elucidate the distinctions between IT2-FLSs and GT2-FLSs.

Next, we provide the mathematical background on MJ-GT2-FLSs, how they are implemented through the α-plane representation, and we discuss the potential issue that we encounter with the MJ-GT2-FLSs: (1) Direct dependency of the SMF shape on the PMF. Inherently, we believe that this dependency affects the learning performance of GT2-FLSs. (2) Curse of dimensionality problem (i.e., rule firing strength). In high-dimensional datasets, rule-firing strengths approach zero, causing learning performance to decrease. Afterward, we introduce Z-GT2-FLSs based on Z-GT2-FSs, their mathematical foundations, i.e., PMF and SMF definitions with T1-FSs. Additionally, we propose solutions for problems in MJ-GT2-FLSs: (1) the SMF shape does not depend on the PMF shape. The mathematical structure of Z-GT2-FSs allows us to eliminate the dependency of the SMF on the PMF. This design choice offers greater flexibility compared to MJ-GT2-FLSs, enabling more adaptable uncertainty modeling. (2) To address the curse of dimensionality problem, we introduce dynamic adjustments to the PMF based on input dimensions. Lastly, for learning of T2-FLSs with unconstrained DL optimizers, we give details of the parameterization tricks we implement for each FLS.

We introduce a dual-focused deep learning (DL) framework for Z-GT2-FLSs, designed to improve both prediction accuracy and uncertainty quantification (UQ). To achieve this, we define a composite loss function that balances these objectives. Using the α-plane representation of GT2-FSs, we construct two separate loss functions tailored for this dual goal. We demonstrate that Z-GT2-FLSs outperform MJ-GT2-FLSs and IT2 counterparts on high-dimensional datasets. To provide a detailed statistical comparison, we include notch and whisker plots for each FLS across all datasets. Furthermore, we study the effect of the number of fuzzy rules P by retraining models for $P = 5$ and $P = 10$. We also assess each model's computational cost in terms of memory usage and training time.

In the last chapter of this thesis, we present a DL framework with Z-GT2-FLSs for learning inverse cumulative distribution by estimating all quantile levels simultaneously. The dual-focused Z-GT2-FLS learning approach is designated for a given confidence level, requiring to undergo retraining for any desired confidence level. In this chapter, we assign α-planes of GT2-FSs with quantile levels, $\tau \in [0, 1]$, (i.e., $\alpha = \tau$). Thus $\alpha \in [0, 1]$ is an input argument to GT2-FLSs instead of being a structural parameter. This structure helps us to design a DL framework for GT2-FLSs to learn predictive distribution. In this context, each α-IT2-FLS learns a quantile level, τ, selected randomly from the uniform distribution $\tau \sim U[0, 1]$ of SQR, (i.e., $\alpha = \tau$), and the model is called Z-GT2-SQR. Building on this, we propose a novel approach called ASQR, which targets the miscalibrated regions identified during validation and selects quantile levels specifically from those regions during training, and the model is called Z-GT2-ASQR. This strategy allows us to capture the full spectrum of quantile levels when compared to the SQR method. Furthermore, we showcase the learning performance of Z-GT2-SQR and Z-GT2-ASQR compared to state-of-the-art DL models for UQ over benchmark datasets. We conclude that Z-GT2-ASQR shows excibit performance compared to state-of-the-art DL UQ methods. We also present Q-Q plots, serving as a means of comparing expected quantile levels against observed ones. These plots help us to understand that ASQR is a versatile approach for capturing the full spectrum of quantile levels.

We believe that the results of this thesis study demonstrate that Z-GT2-FLSs can be a significant tool for fields requiring both accurate predictions and reliable uncertainty estimates, such as healthcare (e.g., diagnostic systems), finance (e.g., risk assessment), and autonomous technologies (e.g., decision-making under uncertainty). The key limitation is the significant computational cost associated with the Z-GT2-FLS framework, particularly impacting training duration for both learning frameworks (dual-focused and distribution estimation). However, a key distinction is that the inference or application time is significantly shorter once the model is trained. This trade-off which is long training period for rapid and reliable inference is justifiable in high-stakes applications such as real-time diagnostics, algorithmic trading, and autonomous systems. In our future work, we will explore the application of Z-GT2-FLSs in time series prediction and real-world scenarios.

REFERENCES

[1] **Xue, G.**, **Wang, J.**, **Zhang, K.**, **and Pal, N.R.** (2023). High-dimensional fuzzy inference systems, *IEEE Trans. Syst. Man Cybern. Syst.*

[2] **Krizhevsky, A.**, **Sutskever, I.**, **and Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *In Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 25. Curran Associates, Inc.

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

[4] **Vaswani, A.**, **Shazeer, N.**, **Parmar, N.**, **Uszkoreit, J.**, **Jones, L.**, **Gomez, A.N.**, **Kaiser, Ł.**, **and Polosukhin, I.** (2017). Attention is all you need, *In Advances in neural information processing systems*, Vol. 30.

[5] **Devlin, J.**, **Chang, M.W.**, **Lee, K.**, **and Toutanova, K.** (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*.

[6] **Levine, S.**, **Finn, C.**, **Darrell, T.**, **and Abbeel, P.** (2016). End-to-end training of deep visuomotor policies, *In The 33rd International Conference on Machine Learning*.

[7] **Abdar, M. and others**. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges, *Inf. Fusion.*, *76*, 243–297.

[8] **Papadopoulos, H.**, **Vovk, V.**, **and Gammerman, A.** (2011). Regression conformal prediction with nearest neighbours, *J. Artif. Intell.*, *40*, 815–840.

[9] **Pearce, T.**, **Brintrup, A.**, **Zaki, M.**, **and Neely, A.** (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach, *In Int. Conf. Mach. Learn.*, Vol. 80.

[10] **Quan, H.**, **Srinivasan, D.**, **and Khosravi, A.** (2014). Short-term load and wind power forecasting using neural network-based prediction intervals, *IEEE Trans. Neur. Net. Learn. Syst.*, *25*(2), 303–315, doi : 10.1109/TNNLS.2013.2276053.

[11] **Chung, Y.**, **Neiswanger, W.**, **Char, I.**, **and Schneider, J.** (2021). Beyond pinball loss: Quantile methods for calibrated uncertainty quantification, *Adv. Neural Inf. Process. Syst.*, *34*, 10971–10984.

[12] **Tagasovska, N. and Lopez-Paz, D.** (2019). Single-model uncertainties for deep learning, *Adv. Neural Inf. Process. Syst.*, *32*.

[13] **Zhao, S.**, **Ma, T.**, **and Ermon, S.** (2020). Individual calibration with randomized forecasting, *In Int. Conf. Mach. Learn.*

[14] **Mendel, J.M.** (2017). *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*. Springer, New York, NY, USA.

[15] **L.A. Zadeh**. (1975). The concept of a linguistic variable and its application to approximate reasoning—i, *Inf. Sci.*, *8*(3), 199–249, doi : https://doi.org/10.1016/0020-0255(75)90036-5.

[16] **L.A. Zadeh**. (1971). Quantitative fuzzy semantics, *Inf. Sci.*, *3*(2), 159–176, doi : https://doi.org/10.1016/S0020-0255(71)80004-X.

[17] **Mendel, J. and John, R.** (2002). Type-2 fuzzy sets made simple, *IEEE Transactions on Fuzzy Systems*, *10*(2), 117–127, doi : 10.1109/91.995115.

[18] **Wagner, C. and Hagras, H.** (2010). Toward general type-2 fuzzy logic systems based on zslices, *IEEE Transactions on Fuzzy Systems*, *18*(4), 637–660, doi : 10.1109/TFUZZ.2010.2045386.

[19] **Shihabudheen, K. and Pillai, G.N.** (2018). Recent advances in neuro-fuzzy system: A survey, *Knowl Based Syst.*, *152*, 136–162.

[20] **Zheng, Y.**, **Xu, Z.**, **and Wang, X.** (2021). The fusion of deep learning and fuzzy systems: A state-of-the-art survey, *IEEE Transactions on Fuzzy Systems*, *30*(8), 2783–2799.

[21] **Tsai, S.H. and Chen, Y.W.** (2022). A novel interval type-2 fuzzy system identification method based on the modified fuzzy c-regression model, *IEEE Trans. Cybern.*, *52*(9), 9834–9845.

[22] **Wiktorowicz, K.** (2023). T2RFIS: type-2 regression-based fuzzy inference system, *Neural Comput. Appl.*, *35*(27), 20299–20317.

[23] **Tavoosi, J.**, **Mohammadzadeh, A.**, **and Jermsittiparsert, K.** (2021). A review on type-2 fuzzy neural networks for system identification, *Soft Computing*, *25*, 7197–7212.

[24] **Han, H.**, **Liu, Z.**, **Liu, H.**, **Qiao, J.**, **and Chen, C.P.** (2021). Type-2 fuzzy broad learning system, *IEEE Trans. Cybern.*, *52*(10), 10352–10363.

[25] **Castro, J.R.**, **Castillo, O.**, **Melin, P.**, **and Rodríguez-Díaz, A.** (2009). A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, *Information Sciences*, *179*(13), 2175–2193.

[26] **Han, H.**, **Liu, Z.**, **Liu, H.**, **Qiao, J.**, **and Chen, C.L.P.** (2022). Type-2 fuzzy broad learning system, *IEEE Trans. Cybern.*, *52*(10), 10352–10363.

[27] **Denœux, T.** (2023). Quantifying prediction uncertainty in regression using random fuzzy sets: The ennreg model, *IEEE Transactions on Fuzzy Systems*, *31*(10), 3690–3699.

[28] **Fumanal-Idocin, J.**, **Takáč, Z.**, **Horanská, L.**, **da Cruz Asmus, T.**, **Dimuro, G.**, **Vidaurre, C.**, **Fernandez, J.**, **and Bustince, H.** (2022). A generalization of the sugeno integral to aggregate interval-valued data: an application to brain computer interface and social network analysis, *Fuzzy Sets and Systems*, *451*, 320–341.

[29] **Beke, A. and Kumbasar, T.** (2023). More than accuracy: A composite learning framework for interval type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems*, *31*(3), 734–744, doi : 10.1109/TFUZZ.2022.3188920.

[30] **Avcı, B.**, **Beke, A.**, **and Kumbasar, T.** (2023). Towards reliable uncertainty quantification and high precision with general type-2 fuzzy systems, *In IEEE Int. Conf. Fuzzy Syst.*, doi : 10.1109/FUZZ52849.2023.10309730.

[31] **Rodrigues, F. and Pereira, F.C.** (2020). Beyond expectation: Deep joint mean and quantile regression for spatiotemporal problems, *IEEE Trans. Neural Netw. Learn. Syst.*, *31*(12), 5377–5389.

[32] **Karnik, N. and Mendel, J.** (1998). Introduction to type-2 fuzzy logic systems, *In IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228)*, Vol. 2, pp. 915–920 vol.2, doi : 10.1109/FUZZY.1998.686240.

[33] **Mendel, J.M.**, **John, R.I.**, **and Liu, F.** (2006). Interval type-2 fuzzy logic systems made simple, *IEEE Transactions on Fuzzy Systems*, *14*(6), 808–821.

[34] **van Krieken, E.**, **Acar, E.**, **and van Harmelen, F.** (2022). Analyzing differentiable fuzzy logic operators, *Artificial Intelligence*, *302*, 103602.

[35] **Kumbasar, T.** (2017). Revisiting karnik–mendel algorithms in the framework of linear fractional programming, *International Journal of Approximate Reasoning*, *82*, 1–21.

[36] **Köklü, A.**, **Güven, Y.**, **and Kumbasar, T.** (2024). Efficient learning of fuzzy logic systems for large-scale data using deep learning, *In International Conference on Intelligent and Fuzzy Systems*, pp. 406–413. Springer.

[37] **Sakalli, A.**, **Kumbasar, T.**, **and Mendel, J.M.** (2021). Towards systematic design of general type-2 fuzzy logic controllers: Analysis, interpretation, and tuning, *IEEE Transactions on Fuzzy Systems*, *29*(2), 226–239, doi : 10.1109/TFUZZ.2020.3016034.

[38] **Bustince, H. and others**. (2016). A historical account of types of fuzzy sets and their relationships, *IEEE Transactions on Fuzzy Systems*, *24*(1), 179–194, doi : 10.1109/TFUZZ.2015.2451692.

[39] **Cui, Y.**, **Wu, D.**, **and Xu, Y.** (2021). Curse of dimensionality for tsk fuzzy neural networks: Explanation and solutions, *In Proc. Int. Jt. Conf. Neural Netw.*

[40] **Köklü, A.**, **Güven, Y.**, **and Kumbasar, T.** (2024). Enhancing interval type-2 fuzzy logic systems: Learning for precision and prediction intervals, *In IEEE Int. Conf. Fuzzy Syst.*

[41] **Güven, Y.**, **Köklü, A.**, **and Kumbasar, T.** (2025). Exploring zadeh's general type-2 fuzzy logic systems for uncertainty quantification, *IEEE Transactions on Fuzzy Systems*, *33*(1), 314–324.

[42] **Cui, Y.**, **Xu, Y.**, **Peng, R.**, **and Wu, D.** (2023). Layer normalization for tsk fuzzy system optimization in regression problems, *IEEE Transactions on Fuzzy Systems*, *31*(1), 254–264, doi : 10.1109/TFUZZ.2022.3185464.

[43] **Wu, D.**, **Lin, C.T.**, **and Huang, J.** (2019). Active learning for regression using greedy sampling, *Information Sciences*, *474*, 90–105.

[44] **Partaourides, H. and Chatzis, S.** (2018). Deep learning with t-exponential bayesian kitchen sinks, *Expert Syst. Appl.*, *98*, 84–92.

[45] **Liu, C.**, **Tang, L.**, **and Liu, J.** (2019). Least squares support vector machine with self-organizing multiple kernel learning and sparsity, *Neurocomputing*, *331*, 493–504.

CURRICULUM VITAE

Name SURNAME: Yusuf GÜVEN

EDUCATION:

- **B.Sc.:** 2023, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control and Automation Engineering Department

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Guven, Y.**, Koklu, A., Kumbasar, T. (2024). Exploring Zadeh's General Type-2 Fuzzy Logic Systems for Uncertainty Quantification. *IEEE Transactions on Fuzzy Systems*.
- **Guven, Y.**, Koklu, A., Kumbasar, T. (2024). Zadeh's Type-2 Fuzzy Logic Systems: Precision and High-Quality Prediction Intervals. *IEEE World Congress on Computational Intelligence*.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Guven, Y.**, Kumbasar, T. (2025). Fuzzy Logic Strikes Back: Fuzzy ODEs for Dynamic Modeling and Uncertainty Quantification. *IEEE Transactions on Artificial Intelligence*.
- **Guven, Y.**, Kumbasar, T. (2025). Adapting GT2-FLS for Uncertainty Quantification: A Blueprint Calibration Strategy. *IEEE Conference on Fuzzy Systems*.
- Gokmen, O.B., **Guven, Y.**, Kumbasar, T. (2025). FAME: Introducing Fuzzy Additive Models for Explainable AI. *IEEE Conference on Fuzzy Systems*.
- Koklu, A., **Guven, Y.**, Kumbasar, T. (2024). Odyssey of Interval Type-2 Fuzzy Logic Systems: Learning Strategies for Uncertainty Quantification. *IEEE Transactions on Fuzzy Systems*.
- Koklu, A., **Guven, Y.**, Kumbasar, T. (2024). Enhancing Interval Type-2 Fuzzy Logic Systems: Learning for Precision and Prediction Intervals. *IEEE World Congress on Computational Intelligence*.
- Koklu, A., **Guven, Y.**, Kumbasar, T. (2024). Efficient Learning of Fuzzy Logic Systems for Large-Scale Data using Deep Learning. *International Conference on Intelligent and Fuzzy Systems*.