Quantile Functions in Neural Embedded Optimization

Two-stage Stochastic and Chance-Constrained Optimization

Taehyeon Kwon

Department of Industrial and Manufacturing Engineering
The Pennsylvania State University

Preview

Key Points

1. **Quantile Functions Overview**
   - Introduce the concept and significance.
2. **Applications**
   1. Chance Constraint Optimization
   2. Two-Stage Stochastic Optimization

Quantile Function

Concepts

- **Definition**: *Quantiles* are values that divide a dataset into equal-sized, contiguous subgroups. They are a way to understand the distribution and spread of data.

- **Purpose**: *Quantiles* help in understanding the position of a particular value in a distribution and are commonly used in statistics to summarize data sets.

- **Examples**:

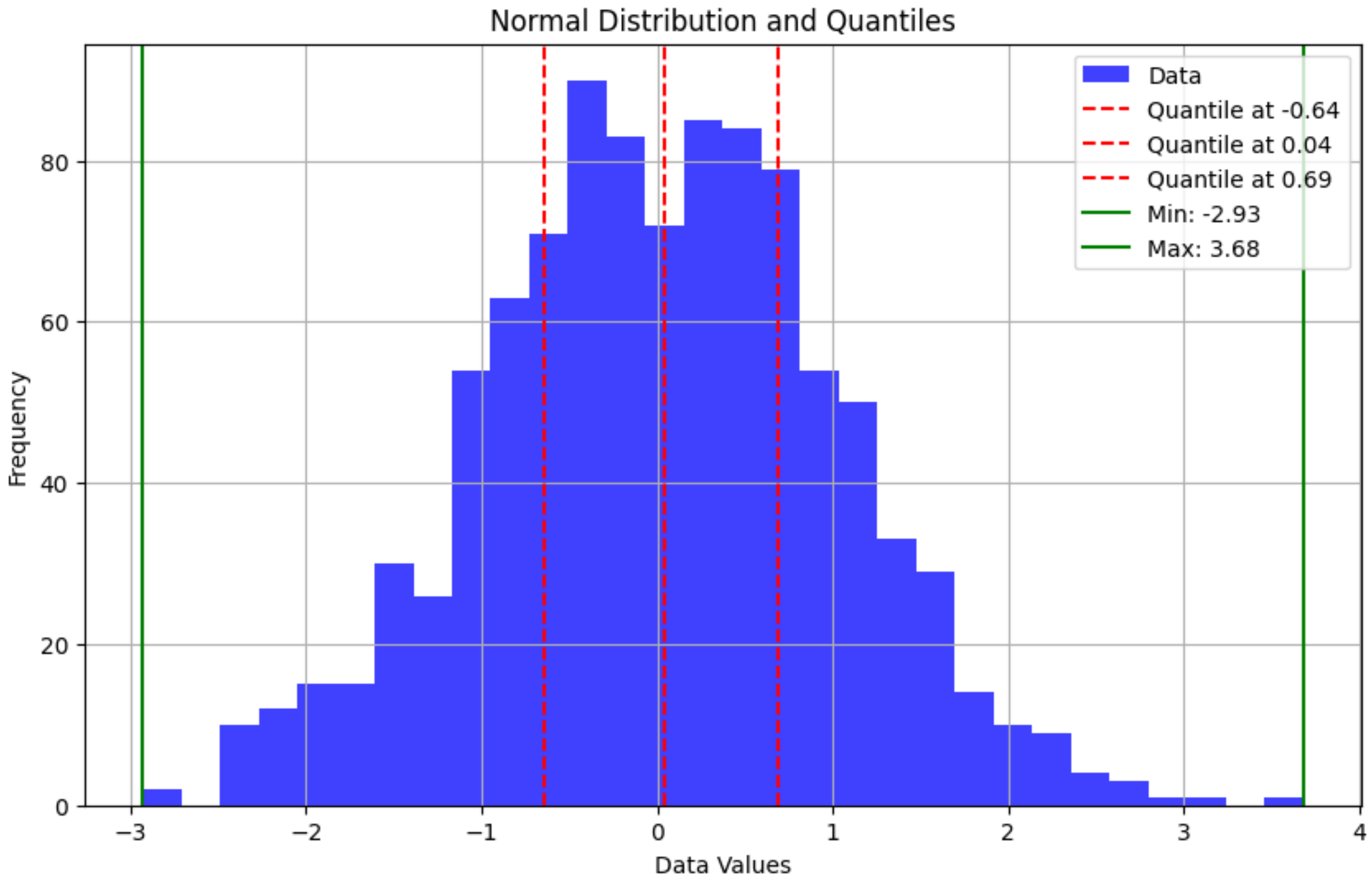
  - Quartiles: Divide data into four equal parts.

  - Percentiles: Divide data into 100 equal parts.

  - Deciles: Divide data into 10 equal parts.

Quantile Function

Concepts



- Randomly sampled 1000 samples from gaussian distribution
- Red dash lines (from left -> right) indicates 25%, 50%, 75% percentiles respectively.

Quantile Function

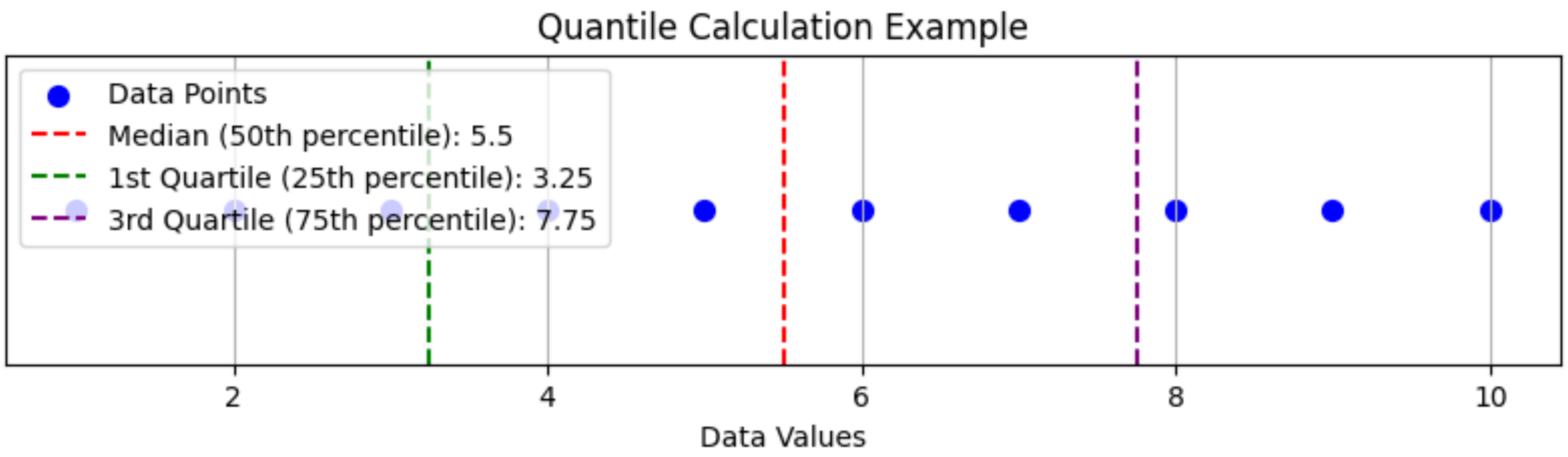
Mathematical Interpretation and Examples

- **Quantile Formula**: A quantile of a dataset divides the data into segments with respect to the cumulative distribution function.

$$p^{th}\text{ quantile} = (x : \Pr(X \leq x) = p)$$

Quantile Function

- **Examples**: Give dataset; [1,2,3,4,5,6,7,8,9,10]
  - First Quartile (25th Percentile): $(2.5 + 3)/2 = 2.75$; meaning 25% of the data is less than or equal to 2.75.
  - Third Quartile (75th Percentile): $(7.5 + 8)/2 = 7.75$; meaning 75% of the data is less than or equal to 7.75.



Quantile Function

Definition

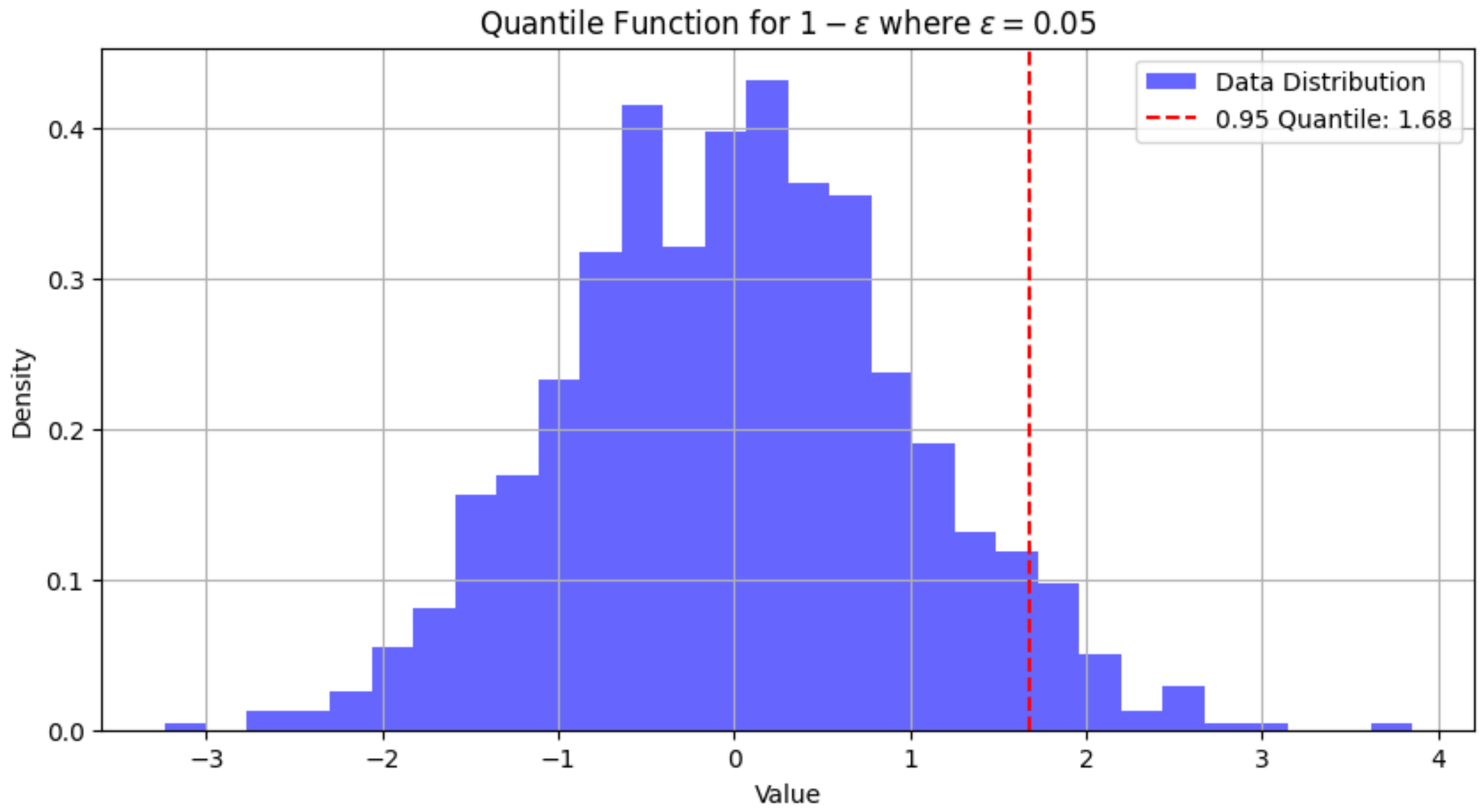
The quantile function $Q^{1-\epsilon}(\cdot)$, with quantile level $1-\epsilon$, where $\epsilon \in [0,1]$ can be defined as follow:

$$\boxed{Q^{1-\epsilon}(X) = \inf\{x \in \mathbb{R} \mid \Pr(X \le x) \ge 1-\epsilon\}}$$

- X : Random variable, $x \in X$
- $\inf\{x : \text{condition of } x\}$: Infimum; the smallest value in the sets which is satisfying conditions for x
  - $\inf\{x \in \mathbb{R} : x > 2\} = 2$ vs $\min\{x \in \mathbb{R} : x > 2\} = ?$
  - $\inf\{x \in \mathbb{R} : x \ge 2\} = 2$ vs $\min\{x \in \mathbb{R} : x \ge 2\} = 2$

Quantile Function

Example



- $X \sim N(0,1) \mid \epsilon = 0.05$
- $Q^{0.95}(X) = \inf\{x \in \mathbb{R} \mid \Pr(X \leq x) \geq 0.95\} = 1.68$

The value **1.68** marks the 95th percentile helps in setting benchmarks, thresholds, or cutoffs in various problems (i.e., finance, quality control, and risk management)

Application: Chance Constraint Optimization

What is Chance Constraint Optimization?

$$\min_{x \in \mathbb{R}^n} \quad c(x) \tag{1}$$

$$\text{s.t.} \quad \Pr\left\{g(x,\xi) \leq \gamma\right\} \geq 1 - \epsilon \tag{2}$$

where the function $g(x, \xi)$ is a function of the decision variables x and the random variables ξ and γ is a threshold for $g(x, \xi)$.

Application: Chance Constraint Optimization

Mathematical Formula: Basic

Let's consider $F(x)$ is the cumulative distribution function (CDF) of probability density function (PDF) $f(x)$

$$F(x) = \Pr\{X \leq x\} \xrightarrow{\text{Extend}} F(x;\gamma) = \Pr\{g(x,\xi) \leq \gamma\}$$

Extension is possible. Since ξ is random variable and $g(x,\xi)$ correspond to random variable X.
Also $\gamma \in \mathbb{R}$ should be a specific value possible set of $g(x,\xi)$

Application: Chance Constraint Optimization

Mathematical Formula: Modification

$$\Pr\left\{g(x,\xi) \leq \gamma\right\} \geq 1-\epsilon \iff F(\gamma, x) \geq 1-\epsilon \tag{3}$$

$$F(\gamma, x) \geq 1-\epsilon \iff Q^{1-\epsilon}(g(x,\xi)) \leq \gamma \tag{4}$$

$$Q^{1-\epsilon}(g(x,\xi)) \leq \gamma \iff \inf\{u \in \mathbb{R} \mid \Pr\{g(x,\xi)\} \leq u) \geq 1-\epsilon\} \leq \gamma \tag{5}$$

Application: Chance Constraint Optimization

Mathematical Formula: Final

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c(x) \\ \text{s.t.} \quad & Q^{1-\epsilon}(g(x,\xi)) \leq \gamma \end{aligned} \tag{6}$$

$Q^{1-\epsilon}(g(x,\xi))$ results specific value $u^* = \inf\{u \in \mathbb{R} \mid \Pr\{g(x,\xi)\} \leq u) \geq 1-\epsilon\}$

Learn the mapping $x \mapsto Q^{1-\epsilon}(g(x,\xi))$ by neural surrogates

Application: Chance Constraint Optimization

Effectiveness

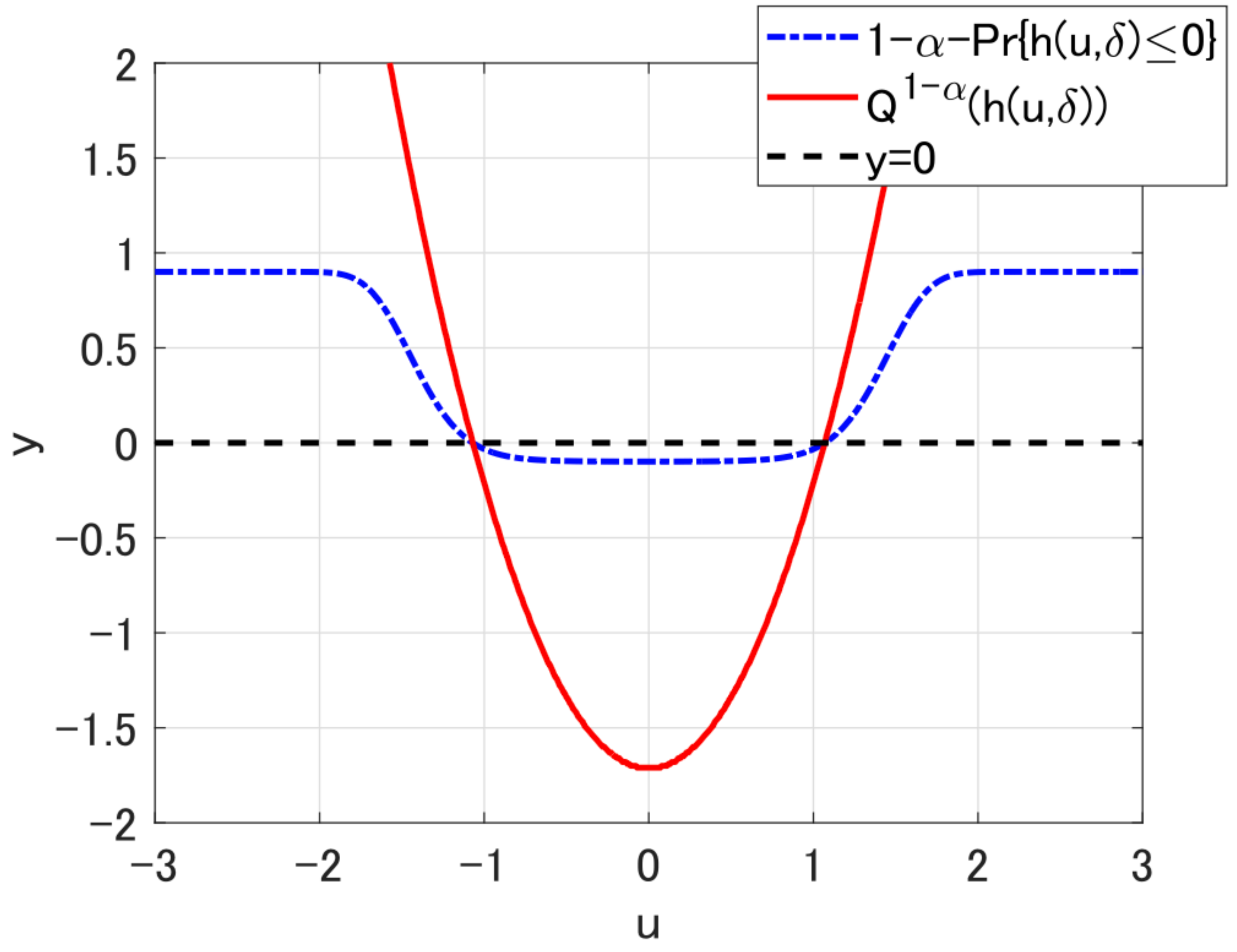


Fig. 1. Comparison of $1-\alpha-\Pr\{h(u,\delta)\leq 0\}$ and $Q^{1-\alpha}(h(u,\delta))$, where $h(u,\delta)=1.5u^2-3+\delta$ and $\delta\sim N(0,1)$ $(\alpha=0.1)$.

Features:

- Quantile function is much less flat
- Feasible region is expanded; not bounced in [0,1]

Shen et al., *IEEE* (2021)

Application: Two-Stage Stochastic Optimization

Overview of Quantile Neural Network Methodology

- **What is QNN?** A QNN is a type of neural network used as a surrogates in two-stage stochastic optimization problems (2SP).

Original 2SP

$$\min_{x\in\mathcal{X}} \; c^\top x + \mathbb{E}_\xi \left[V(x,\xi)\right]$$

$$\text{s.t.} \quad V(x,\xi) := \min_{y\in\mathcal{Y}(x,\xi)} f(y)$$

Learn neural surrogates

Embed QNN into 2SP

$$\min_{x\in\mathcal{X}} \; c^\top x + Q_{NN}(x)$$

$$\text{s.t.} \quad Q_{NN}(x) : x \mapsto Q^{1-\epsilon}(V(x,\xi))$$

Application: Two-Stage Stochastic Optimization

Functional Mechanism of QNN in Optimization

- **Training the QNN:** The network is trained to estimate the "*distribution of the second-stage value*" function based on the first-stage decision variables.
- **Input and Output:**
  - **Input:** First-stage decision variables X
  - **Output:** *Multiple quantiles* to reconstruct the distribution of the second-stage value function.
  - $$Q_{NN}(X) = \{Q^{1-\epsilon^i}(V(X,\xi)) : \epsilon^i \in [0,1], \quad i = 1,\ldots n^L\}$$
- **Network Structure:**
  - QNN uses a multi-output, feed-forward structure.
  - ReLU (Rectified Linear Unit) as activation function.

Application: Two-Stage Stochastic Optimization

QNN Structure for 2SP

Neur2SP

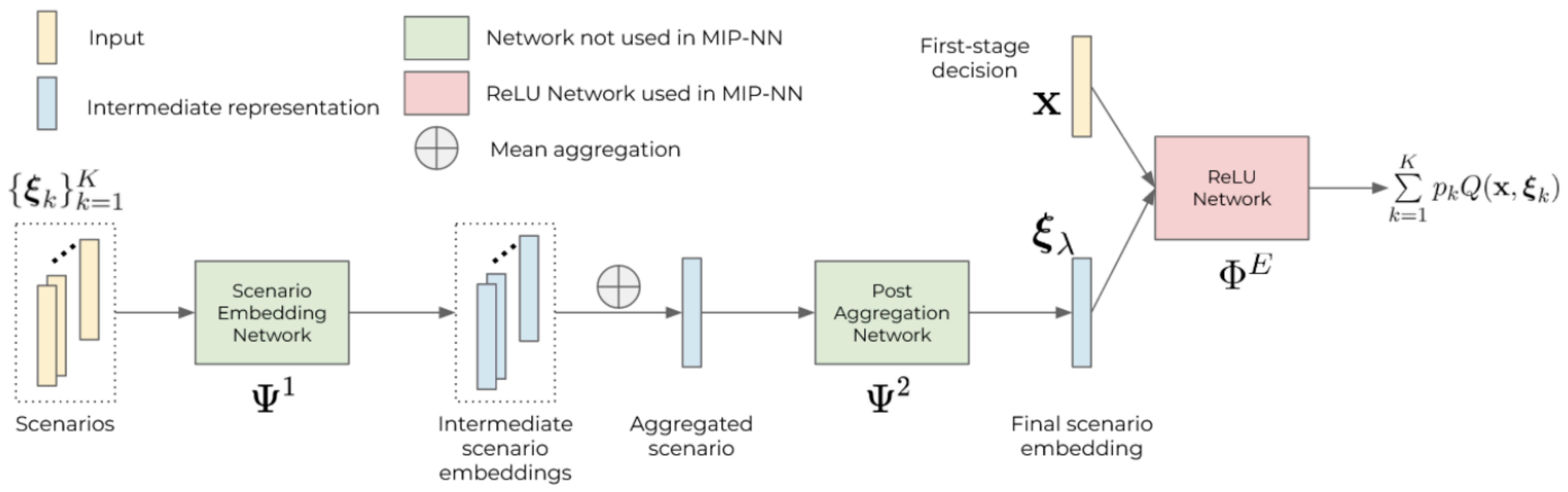


Figure 2: NN-E architecture diagram.

QNN

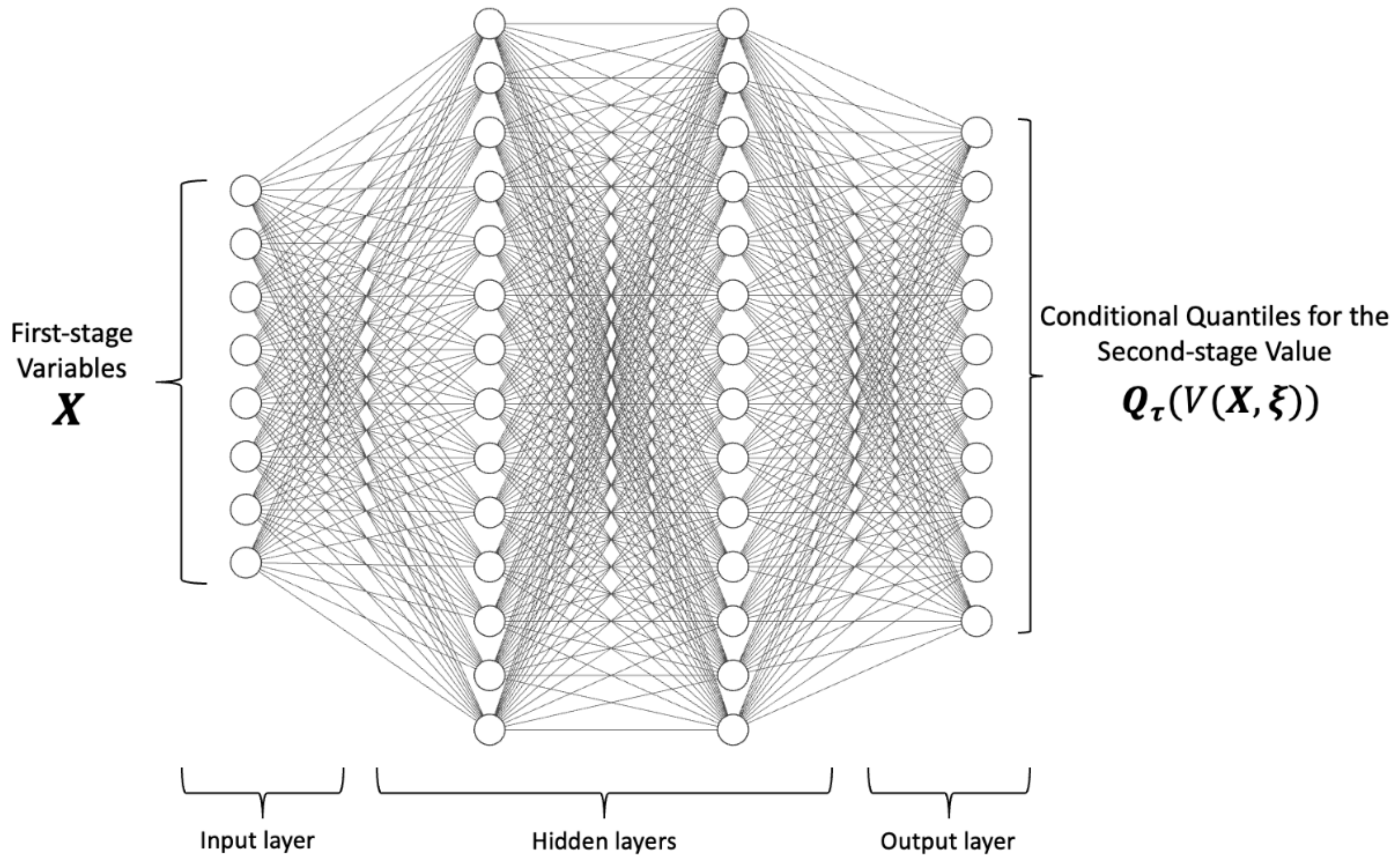


Figure 1: QNN Structure for Two-stage Optimization Problems.

Dumouchelle et al., *NeurIPS* (2022), Alcántara et al., *arXiv* (2024)

Application: Two-Stage Stochastic Optimization

Embedding QNN in Stochastic Optimization

- **Embedding QNN:** Once trained, QNN can be incorporated within the optimization problem as a set of piecewise-linear constraints.
- **Optimization Options:**
  - Optimizing over mean values for expected outcomes.
  - Optimizing over tail mean values for risk-averse strategies.
- **Benefits of Using QNN:**
  - Allows for direct accounting of uncertainty without reliance on multiple scenarios.
  - Provides a robust approach for managing variabilities in second-stage outcomes.

Application: Two-Stage Stochastic Optimization

Performances: computing times

Problem	QNN					IQNN				NN-E	NN-P	SAA
	Data Generation	Training	Tolerance Selection	Problem Solving	**Total Time**	Data Generation	Training	Problem Solving	**Total Time**	**Total Time**	**Total Time**	**Total Time**
CFLP-10-10-100sc										2,490.73 (0.38)	148.99 (8.28)	4,410.60
CFLP-10-10-500sc	38.83	526.43	16.40	3.30	584.96	38.83	407.01	0.04	445.88	2,490.95 (0.60)	347.01 (206.30)	10,800.17
CFLP-10-10-1000sc										2,490.99 (0.64)	997.47 (856.77)	10,800.87
CFLP-25-25-100sc										6,354.50 (0.44)	957.76 (4.86)	10,800.06
CFLP-25-25-500sc	385.14	417.60	47.30	0.32	850.36	385.14	383.92	0.03	769.09	6,354.60 (0.54)	979.31 (26.41)	10,800.14
CFLP-25-25-1000sc										6,354.64 (0.58)	1,007.35 (54.45)	10,800.36
CFLP-50-50-100sc										8,163.28 (1.66)	284.78 (21.10)	10,800.05
CFLP-50-50-500sc	725.31	283.43	207.30	0.32	1,216.36	725.31	258.17	0.03	1,010.51	8,162.87 (1.25)	437.31 (173.63)	10,806.15
CFLP-50-50-1000sc										8,163.06 (1.44)	835.80 (572.12)	10,805.82
IP-I-H-441sc										9,338.79 (0.32)	1,409.06 (1,231.48)	10,800.00
IP-I-H-1681sc	15.46	259.28	4.20	0.06	279.00	15.46	270.00	0.09	285.55	9,338.80 (0.33)	10,994.47 (10,816.89)	10,800.03
IP-I-H-10000sc										9,338.85 (0.38)	—	10,802.10

Table 2: Computing times (in seconds) for the (I)QNN framework and competitive approaches in risk-neutral optimization. NN-E, NN-P, and SAA times are reproduced from Patel et al. (2022). CPU times for solving the resulting optimization problems for NN-E and NN-P are shown in parentheses.

Alcántara et al., *arXiv* (2024)

Application: Two-Stage Stochastic Optimization

Performances: relative gaps QNN vs SAA

Problem	QNN	IQNN	NN-E	NN-P	SAA
CFLP-10-10-100sc	7,129.68 (1.93%)	7,124.11 (1.85%)	7,174.57	7,109.62	**6,994.77**
CFLP-10-10-500sc	7,136.43 (1.90%)	7,114.86 (1.59%)	7,171.79	7,068.91	**7,003.30**
CFLP-10-10-1000sc	7,108.05 (0.27%)	7,095.69 (0.10%)	7,154.60	**7,040.70**	7,088.56
CFLP-25-25-100sc	11,967.66 (0.87%)	11,999.41 (1.13%)	**11,773.01**	**11,773.01**	11,864.83
CFLP-25-25-500sc	11,882.88 (-2.36%)	11,915.06 (-2.10%)	**11,726.34**	**11,726.34**	12,170.67
CFLP-25-25-1000sc	11,870.60 (0.02%)	11,915.35 (0.40%)	**11,709.90**	**11,709.90**	11,868.04
CFLP-50-50-100sc	25,944.94 (2.35%)	27,603.63 (8.89%)	25,236.33	**25,019.64**	25,349.21
CFLP-50-50-500sc	25,906.90 (-7.60%)	27,324.21 (-2.54%)	25,281.13	**24,964.33**	28,037.66
CFLP-50-50-1000sc	25,881.86 (-14.53%)	27,178.49 (-10.25%)	25,247.77	**24,981.70**	30,282.41
IP-I-H-441sc	65.91 (-1.98%)	66.36 (-1.31%)	65.12	65.12	**67.24**
IP-I-H-1681sc	65.60 (0.29%)	**65.74** (0.50%)	65.63	65.34	65.41
IP-I-H-10000sc	**65.89** (1.95%)	65.84 (1.87%)	65.66	—	64.63

Table 3: True objective results for risk-neutral optimization. Relative gaps between QNN and IQNN objective values and the SAA approach are shown in brackets. The best results are highlighted in bold. Results from NN-E, NN-P, and SAA are reproduced from Patel et al. (2022).

Alcántara et al., *arXiv* (2024)