Neural Network Cognitive Diagnosis Method Based on Transfer Learning and Q-Matrix Constraints

Authors: Tao Jinhang, Zhao Wei, Cheng Nuo, Qiao Lifang, Jiang Qiang, Zhao Wei

Date: 2025-11-03T00:00:00+00:00

Abstract

Neural networks, as the most important machine learning methods, have been widely used in cognitive diagnosis, but there is currently no simple yet general neural network cognitive diagnosis method. Therefore, we propose a Q-matrix-constrained neural network cognitive diagnosis method (Bi-QNN), trained using transfer learning. The advantages of the new model are: (1) Practitioners need not specifically design the network architecture, as the new model can adapt to any dataset based on the Q-matrix and interactive Q-matrix; (2) The design principle of the network architecture originates from the GDINA model, enabling it to effectively represent the main effects and interaction effects of attributes; (3) The training scheme based on transfer learning can effectively address the problem of scarce labeled data, improving the model' s usability and applicability. Experimental results demonstrate that: Bi-QNN exhibits lower prediction errors overall on simulated datasets compared to parametric methods GDINA and DINA; within a certain range, the model demonstrates relatively low sensitivity to the number of attributes, and can still maintain relatively good classification accuracy when the number of attributes increases; the transfer learning-based Bi-QNN method can better adapt to datasets with different sample sizes, maintaining superiority over other models under various conditions of both simulated and empirical data; further improvements in model performance are constrained by the use of parametric model-based simulated data, and the model still exhibits some sensitivity to item quality.

Full Text

A Neural Network-Based Cognitive Diagnosis Method via Transfer Learning and Q-Matrix Constraints

TAO Jinhong[1], ZHAO Wei[1], CHENG Nuo[1], QIAO Lifang[2], JIANG Qiang[1]

([1]School of Information Science and Technology, Northeast Normal University, Changchun 130117, China)

([2]College of Education, Hebei Normal University, Shijiazhuang 050000, China)

Abstract

Neural networks, as a cornerstone of machine learning, have been widely applied to cognitive diagnosis. However, a simple and general-purpose neural network approach for cognitive diagnosis remains elusive. This paper proposes a Q-matrix constrained neural network cognitive diagnosis method (Bi-QNN) trained via transfer learning. The advantages of this new model are: (1) Users need not design network architectures manually, as the model can automatically adapt to any dataset based on the Q-matrix and interactive Q-matrix; (2) The network design principle originates from the GDINA model, enabling effective expression of main and interaction effects of attributes; (3) The transfer learning-based training scheme effectively addresses the scarcity of labeled data, enhancing model usability and applicability. Experimental results demonstrate that Bi-QNN achieves lower prediction errors than parametric methods (GDINA and DINA) on simulated datasets overall. Within a certain range, the model exhibits relatively low sensitivity to the number of attributes and maintains satisfactory classification accuracy even as attribute count increases. The transfer learning-based Bi-QNN training method better adapts to datasets of varying sample sizes, maintaining superiority over other models across multiple conditions in both simulated and empirical data. However, further performance improvement is constrained by simulated data based on parametric models, and the model retains some sensitivity to item quality.

Keywords: cognitive diagnosis, Q-matrix, artificial neural network, transfer learning

1. Introduction

Cognitive Diagnostic Assessment (CDA) aims to finely evaluate examinees'latent traits or proficiency levels in specific knowledge and skills based on their response patterns, providing personalized guidance for educational or psychological interventions. This has led to its widespread application in psychological assessment and personalized learning \cite{Xin_{{{et}}}*{{al}}*{2022}}. Researchers have proposed numerous cognitive diagnosis models for both subjective and objective items, which can be categorized into dichotomous and polytomous scoring models \cite{Gao_{{{et}}}*{{al}}*{2021}}, and computationally into para-

metric and nonparametric models \cite{Liu_{{{et}}{{al}}}{2022}}. Among widely applied parametric polytomous models are the Partial Credit DINA model (PC-DINA; \cite{de_{{{la}}{{Torre}}}{2010}}) and the sequential GDINA model (seqGDINA; \cite{Ma_{{{de}}{{la}}}{{{Torre}}{{2016}}}}). *This study focuses on dichotomous scoring models, with representative non-compensatory models including NIDA (noisy inputs, deterministic "and" gate; \cite{Junker{{{Sijtsma}}{{2001}}}}) and DINA (Deterministic Input, Noisy "and" gate; \cite{de{{{la}}{{Torre}}}{2009}})*, while compensatory models include DINO (Deterministic Input, Noisy "or" gate) and NIDO (Noisy Input, deterministic- "or" -gate) proposed by \cite{Templin_{{{Henson}}{{2006}}}}. *Additionally, more generalized models such as GDINA (Generalized DINA; \cite{de{{{la}}{{Torre}}}{2011}})* exist.

Parametric cognitive diagnosis models are based on probability statistics and employ various parameter estimation methods such as maximum likelihood estimation \cite{Sorrel_{{{et}}{{al}}}{2023}}. These methods often encounter boundary problems in parameter estimation when sample sizes are insufficient, leading to overestimated or underestimated model credibility \cite{Yamaguchi_{2023}}. To address these issues, nonparametric cognitive diagnosis methods have been proposed as alternatives for small-class teaching scenarios \cite{Wang_{{{et}}{{al}}}{2021}}. Nonparametric methods are typically divided into vector similarity distance-based approaches and machine learning-based approaches \cite{Guo_{{{Zhou}}{{2021}}}}. *Representative vector distance methods include NPC (Nonparametric Classification Method; \cite{Chiu{{{Douglas}}{{2013}}}}) and GNPC (Generalized Nonparametric Classification Method; \cite{Chiu{{{et}}{{al}}}{2018}})*, along with various improved methods based on Hamming distance, Mahalanobis distance, Manhattan distance, and other vector similarity calculations \cite{Xu_{{{et}}{{al}}}{2023}}.

Among machine learning approaches applied to cognitive diagnosis, clustering-based methods are most prevalent \cite{Guo_{{{et}}{{al}}}{2020}}. Early work by \cite{Chiu_{{{et}}{{al}}}{2009}} used hierarchical agglomerative clustering and K-means to group examinees into clusters with identical attribute patterns. \cite{Kang_{{{et}}{{al}}}{2015}} proposed the Grade Response Clustering Diagnosis Method (GRCDM) using attribute composite score vectors with K-means clustering, later introducing KNN CDM to address low clustering accuracy with small sample sizes \cite{Kang_{{{et}}{{al}}}{2019}}. \cite{Guo_{{{et}}{{al}}}{2020}} proposed a spectral clustering method to overcome K-means limitations. \cite{Zhang_{{{et}}{{al}}}{2023}} incorporated the number of ideal response patterns as the K value to propose EW-KNN, a nonparametric polytomous cognitive diagnosis method. Beyond clustering, researchers have applied other machine learning methods, such as \cite{Liu_{{{Cheng}}{{2018}}}} *who used Support Vector Machines (SVM) for cognitive diagnosis, achieving comparable performance under small sample conditions. With AI development, deep neural network methods have become highly attractive \cite{Liu*{2021}}. For instance,

\cite{Cui_{{{et}}{{al}}}{2016}} combined DINA model ideal response data to train neural network-based cognitive diagnosis models, though results showed unsatisfactory diagnostic performance. \cite{Wang_{{{et}}{{al}}}{2016}} combined Probabilistic Neural Networks (PNN) with SVM for cognitive diagnosis, demonstrating better PNN performance under independent attribute structures. \cite{Chen_{{{Yan}}{{2021}}}} *used DINA-based simulated data to train neural network cognitive diagnosis models for different attribute hierarchical relationships, showing that neural networks can effectively classify attribute mastery patterns while indicating that classification accuracy decreases with increasing attribute structure complexity.* \cite{Nie{{{et}}{{al}}}{2021}} analyzed neural network cognitive diagnosis performance across multiple factors including attribute count, hierarchy, item quality, and sample size, finding that sample size had less impact than item quality and attribute count. Additionally, \cite{Wen_{{{et}}{{al}}}{2020}} used artificial neural networks as measurement models for Hidden Markov Models (HMM) to achieve longitudinal cognitive diagnosis for monitoring student cognitive attribute development. \cite{Wang_{{{et}}{{al}}}{2020}} proposed a scalable neural cognitive diagnosis framework (NeuralCD) that effectively utilizes interaction information between learners and items to obtain interpretable diagnostic results. \cite{Xue_{{{Bradshaw}}_{{2021}}}} integrated neural networks with DINA and DINO models to implement a semi-supervised neural network cognitive diagnosis model to address labeled data scarcity, demonstrating effectiveness while showing that diagnostic accuracy decreases with reduced item discrimination.

In summary, beyond traditional parametric, nonparametric, and clustering-based methods, numerous researchers have applied neural networks to cognitive diagnosis and proven their effectiveness. However, these studies were conducted under specific experimental conditions with particular contextual constraints, leaving no simple and universally applicable neural network cognitive diagnosis method. In practical testing situations, each assessment may involve different numbers of examinees, knowledge elements or attributes, and items. When applying neural network methods to cognitive diagnosis, designing the network structure—including depth and neuron counts per layer—remains a perplexing question, especially challenging for psychology and education scholars without deep AI expertise. This problem has lacked satisfactory solutions. Moreover, neural network parameter estimation falls under supervised learning, typically solved via gradient descent. Training a neural network cognitive diagnosis model with good generalization capability using only small amounts of annotated data has been a persistent challenge \cite{Xue_{{{Bradshaw}}_{{2021}}}}. Consequently, despite neural networks' tremendous success in natural language processing, computer vision, and other domains, they have not been as widely applied and promoted in cognitive diagnosis as parametric probability-based methods or nonparametric vector distance methods.

Therefore, drawing from GDINA model assumptions, we propose a neural network cognitive diagnosis method called Bi-QNN to address current challenges

in model design, training difficulties, and generalizability when applying neural networks to cognitive diagnosis. Specifically, Bi-QNN' s network structure is constrained by both the Q-matrix and an interactive Q-matrix to express main and interaction effects of attributes. This design enables Bi-QNN' s network architecture to automatically adapt to different datasets based on the Q-matrix. More importantly, when using Bi-QNN, educators only need to provide the Q-matrix and attribute interaction relationship matrix to automatically complete model construction, eliminating the difficulty of personally designing neural network architectures. Additionally, to address Bi-QNN training across various scenarios, we design a training scheme based on transfer learning and evaluate Bi-QNN' s performance through extensive simulation experiments and empirical data analysis. This paper is structured as follows: First, we introduce concepts and theories related to the Q-matrix and GDINA model. Second, we elaborate on the construction process of the Bi-QNN cognitive diagnosis model and its training method based on transfer learning. Subsequently, we evaluate Bi-QNN' s performance through simulation and empirical studies. Finally, we discuss results and future directions.

2. Theoretical Foundation

2.1 Q-Matrix

The Q-matrix is a binary matrix describing the relationship between test items and attributes \cite{Tatsuoka_{1995}}, where each row represents an item and each column represents an attribute, defined as in formula (1):

$$\mathbf{Q} = (q_{jk})$$

Formula (1) shows the Q-matrix representing relationships between J items and K attributes, where attributes may refer to knowledge, skills, or latent traits depending on context. For consistency, we use "attribute" throughout. In the Q-matrix, $q_{jk} = 1$ indicates that item j measures attribute k, while $q_{jk} = 0$ indicates it does not. Attribute relationships can be independent or interrelated: independent attributes are mutually unrelated, while interrelated attributes have direct or indirect dependencies. These relationships are typically represented by a reachability matrix $\mathbf{R}$, defined as in formula (2):

$$\mathbf{R} = (r_{K_iKj})$$

where each row of the reachability matrix represents direct or indirect relationships between attribute K_i and other attributes. If $r_{K_iKj} = 1$, attributes K_i and K_j are interrelated; otherwise, they are independent.

2.2 GDINA Cognitive Diagnosis Model

Our neural network design is inspired by the GDINA model, whose item response calculation is shown in formula (3):

$$P_j(\alpha_\ell) = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{jk}\alpha_{\ell k} + \sum_{k<k'}^{K_j^*} \delta_{jkk'}\alpha_{\ell k}\alpha_{\ell k'} + \cdots + \delta_{j1\cdots K_j^*} \prod_{k=1}^{K_j^*} \alpha_{\ell k}$$

The GDINA model reduces examinees' attribute mastery patterns on item j to $L = 2^{K_j^*}$ types based on the actual attributes measured by each item, where K_j^* represents the number of attributes actually measured by item j, calculated as in formula (4):

$$K_j^* = \sum_{k=1}^{K} q_{jk}$$

where $\alpha_{\ell j}^*$ is an $L \times K^*$ binary matrix representing the collection of attribute patterns composed of attributes actually measured by item j.

As shown in formula (3), the GDINA model decomposes the contribution of various attributes to correctly answering an item into three components: First, the intercept δ_{j0} represents the probability of correctly answering item j without mastering any attributes; second, the main effects δ_{jk} represent the direct contribution of mastering attribute k to correctly answering item j; the remaining terms are interaction effects, where $\delta_{jkk'}$ represents the indirect contribution of simultaneously mastering attributes k and k' to correctly answering item j, and $\delta_{j1,\ldots,K_j^*}$ represents the contribution of mastering all attributes measured by item j to correctly answering it.

3. Bi-QNN Model

3.1 Interaction Relationship Matrix and Interactive Q-Matrix

For the Q-matrix shown in formula (1), according to the binomial theorem and excluding self-relationships, there are $2^K - K - 1$ interaction relationships among K attributes. This clearly does not match the actual interrelationships among attributes in real assessments. To represent attribute relationships more concisely and realistically, this paper uses a binary matrix called the interaction relationship matrix $\mathbf{Q}^{\#}$ to represent interrelationships among multiple attributes, as shown in formula (5):

$$\mathbf{Q}^{\#} = (q_{mk}^{\#})$$

The interaction relationship matrix $\mathbf{Q}^{\#}$ consists of M rows and K columns, where each row represents one type of interaction relationship among attributes.

For example, if $\mathbf{Q}^{\#}$ contains a row $\mathbf{q}^{\#} = (1, 0, 1, 1)$ with four attributes, it indicates an interaction among the first, third, and fourth attributes.

The interaction relationship matrix $\mathbf{Q}^{\#}$ characterizes existing interaction relationships among attributes but does not represent interactions present in each item. Therefore, we use a binary matrix called the interactive Q-matrix $\mathbf{Q}^{*}$ to represent interactions present in each item of the Q-matrix, obtained by computing the Q-matrix and interaction relationship matrix $\mathbf{Q}^{\#}$ as shown in formula (6):

$$\mathbf{Q}^{*} = (q_{jm}^{*}) = \prod_{k=1}^{K} q_{jk}^{{q_{mk}}^{\#}}, \quad j = 1, 2, \ldots, J, \quad m = 1, 2, \ldots, M$$

where M is the number of rows in interaction relationship matrix $\mathbf{Q}^{\#}$, and $q_{jm}^{*} = 1$ indicates that item j contains the m-th interaction relationship from matrix $\mathbf{Q}^{\#}$, while $q_{jm}^{*} = 0$ indicates it does not.

3.2 Neural Network Cognitive Diagnosis Model Constrained by Q-Matrix and Interactive Q-Matrix

The release of GPT-4 marked further breakthroughs for deep neural networks across multiple domains including image and natural language processing. The earliest deep feedforward neural network was proposed by \cite{Rosenblatt_{1958}}, and \cite{Rumelhart_{1986}}' s BP algorithm solved neural network parameter computation problems, enabling substantial development and remarkable achievements across domains. Therefore, drawing from GDINA model principles and combining deep neural networks, this paper proposes a neural network cognitive diagnosis model constrained by the Q-matrix and interactive Q-matrix, named Bi-QNN.

As formula (3) shows, the GDINA model divides contributions to correct item responses into three major components: the baseline probability (intercept), main effects of individual attributes, and interaction effects among multiple attributes. Inspired by the GDINA model, our Bi-QNN architecture is shown in Figure 1 [Figure 1: see original paper], containing two computation streams: a green stream representing main effects and an orange stream representing interaction effects, with bias terms retained in hidden layer neuron calculations. This design maintains high consistency between Bi-QNN computations and the GDINA model.

The detailed Bi-QNN computation process is as follows. First, the main effects stream has dimensions determined by the Q-matrix. In this stream, the input layer consists of student response patterns with J neurons, each representing an item; the hidden layer comprises K neurons, each representing an attribute. The specific computation is shown in formula (7):

$$\mathbf{M} = \text{ReLU}(\mathbf{X}(\mathbf{Q} \odot \mathbf{W}_m) + \mathbf{b}_m)$$

where $\mathbf{M} \in \mathbb{R}^{N \times K}$ represents the output of the main effects hidden layer, $\mathbf{X} \in \mathbb{R}^{N \times J}$ represents examinees' response data, $\mathbf{W}_m \in \mathbb{R}^{J \times K}$ represents the weights for this hidden layer, and $\mathbf{b}_m \in \mathbb{R}^K$ is the bias term for each hidden neuron. In the GDINA model, main effects are typically considered non-negative, so $\text{ReLU}(\cdot)$ is used as the activation function for main effects. Notably, $\odot$ denotes the Hadamard product; by computing the Hadamard product of the Q-matrix and hidden layer weights, we constrain connections between this hidden layer and the input layer neurons.

The orange stream represents interaction effects among attributes. In the GDINA model, interaction effects for any item are calculated based on the number of actually measured attributes K_j^*, specifically $2^{K_j^*} - K_j^* - 1$ types of interaction effects. However, not all combinations of attributes truly exist and influence each other in practice. Moreover, since binomial expansion terms grow exponentially, the number of interaction effects increases dramatically with attribute count, causing GDINA model parameters to increase sharply and requiring larger sample sizes for reliable parameter estimation. If we designed the neural network model completely according to GDINA' s interaction effect representation, the network structure would become extremely wide and sparse. Overly wide networks increase model complexity, raising computational costs and training time, while excessively sparse networks affect gradient propagation, destabilizing training, hindering convergence, and reducing generalization capability in practical applications.

To overcome these issues, we reference GDINA' s philosophy but do not fully adopt its interaction effect representation. Instead, we introduce the interactive Q-matrix to optimize network structure, reducing unnecessary nodes and connections to ensure computational efficiency while maintaining model performance. Specifically, for interaction relationships among K attributes, we expect experts to explicitly define attribute interactions based on professional knowledge and practical circumstances—that is, the interaction relationship matrix $\mathbf{Q}^{\#}$. Using formula (6), we compute the interactive Q-matrix $\mathbf{Q}^*$ and design the interaction effect stream' s network structure accordingly. In Bi-QNN, interaction effect computation has two hidden layers. The first hidden layer computation is shown in formula (8):

$$\mathbf{I}_1 = \tanh[\mathbf{X}(\mathbf{Q}^* \odot \mathbf{W}_{i1}) + \mathbf{b}_{i1}]$$

where $\mathbf{I}_1 \in \mathbb{R}^{N \times M}$ represents the first hidden layer' s output, $\mathbf{X} \in \mathbb{R}^{N \times J}$ represents examinees' response data, $\mathbf{Q}^* \in \mathbb{R}^{J \times M}$ is the interactive Q-matrix, $\mathbf{W}_{i1} \in \mathbb{R}^{J \times M}$ represents this layer' s weights, and $\mathbf{b}_{i1} \in \mathbb{R}^M$ is the bias term for each neuron. $\tanh(\cdot)$ is the activation function with range $(-1, 1)$. As before, the interactive Q-matrix $\mathbf{Q}^*$ constrains the network structure. The second hidden layer computation is shown in formula (9):

$$\mathbf{I}_2 = \tanh(\mathbf{I}_1\mathbf{W}_{i2} + \mathbf{b}_{i2})$$

where $\mathbf{I}_2 \in \mathbb{R}^{N\times K}$ represents the second hidden layer' s output, $\mathbf{W}_{i2} \in \mathbb{R}^{M\times K}$ represents this layer' s weights, and $\mathbf{b}_{i2} \in \mathbb{R}^K$ is the bias term.

Applying the Sigmoid activation function to the sum of main and interaction effects yields Bi-QNN' s predicted attribute mastery probabilities $\hat{\mathbf{A}}$, as shown in formula (10):

$$\hat{\mathbf{A}} = \sigma((\mathbf{M} + \mathbf{I}_2)\mathbf{W} + b)$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{N\times K}$ represents the model' s predictions of examinee attribute mastery, and $\sigma(\cdot)$ is the Sigmoid activation function. Following parametric models' probability threshold settings, values in $\hat{\mathbf{A}}$ greater than 0.5 are set to 1 and otherwise to 0, yielding discretized attribute mastery patterns.

Formulas (7)-(10) complete Bi-QNN' s forward computation. Model parameters are updated via backpropagation, with Bi-QNN' s loss function being mean squared error, as shown in formula (11):

$$\mathcal{L}(\theta) = \frac{1}{NK}(\hat{\mathbf{A}} - \mathbf{A})^2$$

where θ represents model parameters including all layer weights and biases, $\mathbf{A}$ is examinees' true attribute mastery patterns, and N is the number of examinees.

4. Transfer Learning-Based Training

4.1 Transfer Learning and Method Selection

In machine learning, transfer learning is a specific learning paradigm that aims to transfer knowledge learned from one task to a related task, thereby improving learning effectiveness and generalization capability on the new task \cite{Pan_{{{Yang}}_{{2010}}}}. Transfer learning addresses the lack of sufficient labeled data in new tasks. By transferring knowledge from related tasks, it enhances learning effectiveness and generalization while avoiding the high cost and implementation difficulty of re-collecting and manually annotating data for new tasks. This characteristic directly addresses the challenges of variable scenarios and scarce annotated data in cognitive diagnosis tasks.

For clarity, we formalize transfer learning definitions. First, a domain is the subject of model learning, consisting of data and its probability distribution. A domain can be formally represented as in formula (12):

$$\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, P(\mathbf{x}, y)\}$$

where $\mathcal{D}$ represents a domain, $\mathcal{X}$ and $\mathcal{Y}$ denote feature and label spaces respectively. Any sample $(\mathbf{x}_i, y_i)$ in a domain contains features and corresponding labels, i.e., $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$. Domain samples follow probability distribution $P(\mathbf{x}, y)$, i.e., $(\mathbf{x}, y) \sim P(\mathbf{x}, y)$. Transfer learning typically involves a source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$. The source domain contains abundant labeled data from which the model learns knowledge to be transferred, while the target domain is the recipient of this knowledge. Transfer learning aims to transfer knowledge learned in the source domain to the target domain.

Given a source domain $\mathcal{D}_s = \{\mathbf{x}_i, y_i\}_{i=1}^{N_s}$ and target domain $\mathcal{D}_t = \{x_j, y_j\}_{j=1}^{N_t}$, where $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, transfer learning aims to learn a target prediction function $f : \mathbf{x}_t \mapsto y_t$ using source domain data when at least one of three conditions holds: (1) different feature spaces, $\mathcal{X}_t \neq \mathcal{X}_s$; (2) different label spaces, $\mathcal{Y}_t \neq \mathcal{Y}_s$; (3) same feature and label spaces but different probability distributions, $P_t(\mathbf{x}, y) \neq P_s(\mathbf{x}, y)$. The goal is to minimize prediction error ℓ on the target domain, as shown in formula (13):

$$f^* = \arg\min_f \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_t} \mathcal{L}(f(\mathbf{x}), y)$$

where f^* is the optimal prediction function on the target domain obtained through transfer learning, and $\mathcal{L}(\cdot)$ is the loss function.

Transfer learning can be categorized into instance-based, feature-based, relationship-based, and model-based approaches \cite{Zhuang_{{{et}}}{{al}}}{2021}}. While instance, feature, and relationship-based methods received early attention, they have practical limitations. Instance-based methods heavily depend on distribution similarity between source and target domains, risking negative transfer when domain differences are large \cite{Pan_{{{Yang}}}{{2010}}}. *Feature-based methods face high implementation difficulty in aligning feature spaces while maintaining discriminability \cite{Wang{{{Deng}}}{{2018}}}. Relationship-based methods often rely on modeling complex relationships between source and target tasks, limiting applicability \cite{Zhuang{{{et}}}{{al}}}{2021}}.* In contrast, model-based transfer learning can share parameters learned in the source domain, especially in neural networks where only fine-tuning on the target domain is needed for effective transfer \cite{Tan_{{{et}}}{{al}}}{2018}}. This makes model-based transfer learning more versatile and generalizable for neural network training. This study focuses on neural network-based cognitive diagnosis, facing issues of small sample sizes, high data annotation costs, and imbalanced attribute mastery pattern categories. Therefore, we select model-based transfer learning to optimize Bi-QNN.

4.2 Pre-training and Fine-tuning for Neural Network Cognitive Diagnosis

Neural network training aims to minimize the loss between predicted and true values, requiring pre-annotated ground truth results. In cognitive diagnosis, the goal is automated assessment of student knowledge mastery, not manual diagnosis. However, neural network training itself requires manually diagnosed data as training benchmarks, creating a significant contradiction. Moreover, tests are typically one-time events; each exam usually requires new items, meaning test forms are rarely reused with the same examinees. Even if an excellent neural network model is trained, it only applies to a specific test, and since most tests are one-time, neural network-based cognitive diagnosis models have poor reusability. This explains why neural networks, despite remarkable achievements across domains, remain underutilized in cognitive diagnosis.

This paper addresses this dilemma using model-based transfer learning with pre-training and fine-tuning. Specifically, we pre-train neural network models on large simulated datasets generated by one or multiple parametric cognitive diagnosis models, then transfer the pre-trained models to smaller datasets for fine-tuning, completing neural network cognitive diagnosis training. This process includes two steps: First, pre-training: for a given Q-matrix, generate pre-training simulated data source domain $\mathcal{D}_s$ based on parametric cognitive diagnosis methods, learning a neural network cognitive diagnosis target function f on $\mathcal{D}_s$ to minimize cost ℓ, as shown in formula (14):

$$\theta_s^* = \arg\min_{\theta} \mathcal{L}(\mathcal{D}_s; \theta)$$

where θ represents function f' s parameters, θ_s^* denotes optimal parameters learned on source domain $\mathcal{D}_s$, and $\mathcal{L}$ is the loss function. The second step is fine-tuning: retrain the cognitive diagnosis target function f on the actual prediction dataset target domain $\mathcal{D}_t$ using pre-trained parameters θ_s^*. Specifically, learn an updated cognitive diagnosis target function f characterized by θ_t^* on target domain $\mathcal{D}_t$, as shown in formula (15):

$$\theta_t^* = \arg\min_{\theta} \mathcal{L}(\theta|\theta_s^*, \mathcal{D}_t)$$

where θ_t^* is the optimal parameter for cognitive diagnosis target function f on target domain $\mathcal{D}_t$ with minimal cost. Figure 2 [Figure 2: see original paper] illustrates the specific process of training Bi-QNN via transfer learning in this study.

The process involves first training a source domain network on simulated dataset $\mathcal{D}_s$, then transferring knowledge about main and interaction effects learned in the source domain to the target domain by copying parameters from the source network' s main effect and interaction effect layers to the target network for fine-tuning, where model parameters are retrained.

5. Simulation Study

5.1 Research Purpose

The simulation study aims to comprehensively evaluate the performance of transfer learning-trained Bi-QNN in terms of classification error and accuracy across broader conditions, while comparing its performance with neural network methods (ANN), parametric methods (GDINA, DINA), and nonparametric methods (NPC, GNPC) to discuss respective advantages under different conditions.

5.2 Research Design

To comprehensively evaluate Bi-QNN in simulation experiments, we generate examinee attribute distributions using multivariate normal distributions. Regarding attribute count, existing research indicates typical ranges of 3 to 8 attributes \cite{Qin_{{{Guo}}*{{2023}}}}, *with 3 attributes common in small-sample scenarios and 5 attributes most widely used in simulation studies* *cite{Najera{{{et}}*{{al}}}*{2021}}. Item count is typically determined by the ratio of items to attributes, generally requiring at least 3 times more items than attributes \cite{Song_{{{et}}*{{al}}}*{2024}}. To better assess model sensitivity to attribute and item counts, we designed three Q-matrix specifications with (attribute count, item count) pairs of (3, 15), (5, 15), and (5, 25). For item quality, we consider both high-quality and low-quality items when generating simulated data. Following previous research, high-quality items are defined as those with guessing and slip parameters drawn from uniform distributions $U(0.05, 0.15)$, while low-quality items have parameters from $U(0.15, 0.30)$ \cite{Cui_{{{et}}*{{al}}}*{2016}, Nie_{{{et}}*{{al}}}*{2021}, Guo_{{{Zhou}}_{{2021}}}}. Additionally, we obtain subsamples with sample sizes N ($N = 50, 100, 200, 300, 500$) under each condition. These settings create $2 \times 2 \times 2 \times 5 = 40$ conditions for generating diverse experimental data.

5.2.1 Item Simulation The Q-matrix for 3 attributes and 15 items in the simulation experiment is designed as in formula (16). $\mathbf{Q}_1$ consists of two identity matrices to ensure completeness and parameter identifiability \cite{Chiu_{2013}, Xu_{{{Zhang}}_{{2016}}}}, with remaining items containing at least 2 attributes to form more complex attribute structures.

The Q-matrix for 5 attributes and 15 items matches that in \cite{Zhan_{{{et}}*{{al}}}*{2022}}, as shown in formula (17):

The Q-matrix for 5 attributes and 25 items references \cite{Zhan_{{{et}}*{{al}}}*{2022}}' s design for 5 attributes and 30 items, adapted for this study as shown in formula (18):

The interaction relationship matrix is generated by randomly selecting a specified number of interactions from the $2^K - K - 1$ possible interactions among K attributes for each experiment, ensuring no duplicate relationships. For the

3-attribute and 5-attribute interactive Q-matrices, we set interaction counts of 4 and 10 respectively, ensuring each interaction involves at least two attributes without repetition. Thus, $\mathbf{Q}_1$' s interaction matrix is a 4×3 binary matrix, while $\mathbf{Q}_2$ and $\mathbf{Q}_3$ are 10×5 binary matrices, representing 4 selected interactions for 3-attribute data and 10 randomly selected interactions from 26 potential interactions for 5-attribute data.

5.2.2 Examinee Simulation To make attribute distributions in the simulation study more realistic, continuous ability value vectors $\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iK})$ for any examinee i on each attribute are generated from multivariate normal distribution $MVN(0, \Sigma)$, where off-diagonal elements of the attribute correlation covariance matrix Σ are set to 0.5. Examinee attribute mastery patterns $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iK})$ are calculated as in formula (19):

$$\alpha_{ik} = \begin{cases} 1 & \text{if } \theta_{ik} \geq \Phi^{-1}\left(\frac{k}{K+1}\right) \\ 0 & \text{otherwise} \end{cases}$$

where K is the number of attributes and $\Phi(\cdot)$ is the normal distribution CDF.

Source domain simulated data consists of 1000 samples stratified from datasets generated by GDINA and DINA under both high- and low-quality conditions. Thus, source domain datasets under different Q-matrices each contain 4000 samples: 1000 high-quality and 1000 low-quality samples from DINA, and 1000 high-quality and 1000 low-quality samples from GDINA. Target domain data uses one-quarter of the source domain sample size, i.e., 1000 samples total, obtained similarly. Sub-datasets and test sets are then sampled according to different sample sizes N. Source domain data is generated only once, with models pre-trained 10 times on the source domain; the model with the smallest and most stable average loss is selected as the pre-trained model. For Bi-QNN retraining, ANN training, and parametric/nonparametric model fitting, target domain data is regenerated each time and sampled according to subsample size N. To reduce random experimental error, following \cite{Nie_{{{et}}*{{al}}}{2021}}, each experiment with different sample sizes is repeated 30 times.

5.2.3 Experimental Procedure

1. Set neural network hyperparameters: For both ANN and Bi-QNN, training iterations are 100; batch size is 16 when sample size $\leq$ 100, otherwise 32; learning rates are 0.001 for datasets generated from $\mathbf{Q}_1$ and $\mathbf{Q}_2$, and 0.002 for $\mathbf{Q}_3$. ANN architecture follows \cite{Cui_{{{et}}*{{al}}}{2016}} and \cite{Nie_{{{et}}*{{al}}}{2021}}, with a 3-layer structure (input, hidden, output) where the hidden layer matches \cite{Cui_{{{et}}*{{al}}}{2016}}' s design.
2. Pre-train Bi-QNN on each condition' s source domain simulated dataset, where the interactive Q-matrix determining interaction effect neuron

counts is computed using formula (6).

3. Perform transfer of the pre-trained Bi-QNN model to the target domain and train ANN: sample sub-training and test sets from the target domain dataset, then fine-tune Bi-QNN and train ANN on the training set.
4. Use trained Bi-QNN and ANN models to predict on the test set, simultaneously fit parametric and nonparametric models on the test set, and finally calculate each model' s performance across different metrics.

5.3 Evaluation Metrics

Root Mean Square Error (RMSE) is selected to evaluate prediction error between model predictions and true values, defined as in formula (20):

$$\mathrm{RMSE} = \sqrt{\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}(\hat{p}_{ik} - \alpha_{ik})^2}$$

where N is sample size, K is attribute count, $\alpha_{ik} \in \{0, 1\}$ represents examinee i' s true attribute mastery on attribute k, and $\hat{p}_{ik}$ represents the model-predicted mastery probability.

To comprehensively evaluate Bi-QNN' s classification accuracy for examinee attribute mastery, Attribute Match Ratio (AMR) and Pattern Match Ratio (PMR) are used as performance metrics \cite{Wang_{{{et}}{{al}}}{2015}}. AMR and PMR are defined in formulas (21) and (22):

$$\mathrm{AMR} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{I}(\hat{\alpha}_{ik} = \alpha_{ik})$$

where AMR represents consistency between predicted and true attributes, $\mathbb{I}(\cdot)$ is the indicator function, and $\hat{\alpha}_{ik} \in \{0, 1\}$ represents model-predicted attribute mastery.

$$\mathrm{PMR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{\alpha}_{i} = \alpha_{i})$$

where PMR represents consistency between predicted and true examinee attribute mastery patterns, with $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK})$ and $\hat{\alpha}_i = (\hat{\alpha}_{i1}, \hat{\alpha}_{i2}, \ldots, \hat{\alpha}_{iK})$ representing true and predicted attribute mastery patterns for examinee i.

5.4 Results

5.4.1 RMSE Results Table 1 presents average RMSE results across different conditions. From the item quality dimension, all models show higher RMSE scores on low-quality datasets than high-quality datasets under the same conditions, indicating that all models' predictive performance is affected by data quality. Bi-QNN shows smaller impact magnitude under the same conditions, demonstrating lower sensitivity to data quality. From the sample size perspective, all models' RMSE scores generally decrease with increasing sample size under the same conditions, showing that sample size affects model performance. However, Bi-QNN' s RMSE trend with sample size is relatively moderate, indicating better robustness.

Furthermore, Bi-QNN achieves lower RMSE scores than other models under most conditions, meaning it achieves smaller overall prediction errors and better system stability. Additionally, across simulated datasets SD1, SD2, and SD3, RMSE scores generally follow the trend SD2 > SD3 > SD1, indicating better performance on less complex datasets. Specifically, models perform better with fewer attributes at the same item count, and with more items at the same attribute count. The former can be observed by comparing RMSE scores on SD1 versus SD2, the latter by comparing SD2 versus SD3.

In summary, all models show smaller overall prediction errors on high-quality, larger-sample, less complex datasets. Bi-QNN outperforms all other models across all tested conditions, validating its predictive reliability and stability.

5.4.2 Attribute Classification Results Table 2 presents attribute classification accuracy (AMR) across experimental conditions, showing average values. From the item quality dimension, all models achieve significantly higher AMR scores on high-quality datasets. Parametric and nonparametric models' AMR scores decrease by approximately 0.1 on low-quality datasets, while Bi-QNN and ANN show smaller decreases of about 0.05, indicating stronger robustness to item quality, with Bi-QNN achieving higher overall AMR scores.

From the sample size dimension, all models generally achieve higher AMR scores on larger-sample datasets, showing that larger samples improve stability and accuracy. Notably, ANN' s performance fluctuates more with sample size on low-sample datasets. For example, on low-quality SD3 dataset at sample size 100, both ANN and Bi-QNN show AMR score declines, indicating that increased input neurons with small samples reduce training stability.

Comparing across datasets, all models perform better on SD1 than SD3, and SD3 better than SD2. All models achieve higher AMR scores on SD1 than SD2, showing that reducing attribute count improves performance when item count is constant. This confirms attribute count as an important performance factor, where Bi-QNN demonstrates stronger adaptability and stability. Comparing SD2 and SD3 shows that with constant attribute count, more items improve predictive reliability and performance. Comparing SD1 and SD3 reveals better

performance on less complex datasets even with equal item-to-attribute ratios. The performance improvement from SD2 to SD1 exceeds that from SD2 to SD3, indicating models are more affected by attribute count than item count. Parametric models are more sensitive to attribute count changes, while Bi-QNN shows more stable performance across attribute counts.

In conclusion, model performance is influenced by sample size, item quality, attribute count, and item count, with attribute count having greater impact than item count. High-quality data and large samples improve accuracy. Except for GNPC slightly outperforming Bi-QNN on high-quality SD1 at sample size 50, Bi-QNN achieves superior AMR scores under all other conditions, demonstrating excellent performance and stability across diverse simulation conditions.

5.4.3 Pattern Classification Results Table 3 shows pattern classification accuracy (PMR) across conditions, using stricter evaluation criteria than AMR.

Overall, PMR scores are lower than AMR scores across all models, consistent with past research \cite{Chen_{{{Yan}}{{2021}}}, *Najera*{{{et}}{{al}}}{2021}}. Trends in PMR mirror those in AMR: higher scores on high-quality, larger-sample, lower-attribute-count datasets. For example, PMR scores are higher on SD1 than SD3, and SD3 than SD2, reaffirming that data quality, sample size, and attribute count are important performance factors.

Notably, on high-quality SD1 dataset, GNPC achieves leading PMR performance over all models at sample sizes $\leq$ 200, while Bi-QNN surpasses all models when sample size $>$ 200. This suggests nonparametric models retain advantages in high-quality, low-attribute, small-sample scenarios, but Bi-QNN demonstrates superior performance as sample size increases. Bi-QNN shows comprehensive advantages across most conditions, particularly on low-quality datasets, where it leads other models by approximately 15-20% in PMR and 5-10% in AMR, indicating stronger robustness and more effective capture of latent cognitive patterns in low-quality data.

In summary, PMR metrics validate models' adaptability under stricter classification standards and highlight Bi-QNN' s performance in complex data scenarios, showing strong stability and generalization capability in both high- and low-quality datasets.

6. Empirical Study

6.1 Data and Experimental Setup

To further validate transfer learning-based Bi-QNN training effectiveness, we use the widely adopted fraction subtraction dataset in cognitive diagnosis research. The dataset contains 536 response samples. We use both the original dataset with 20 items and 8 attributes (FRAC) and a reduced subset with 15 items and 5 attributes (Sub FRAC), available from the R package CDM. The former' s

Q-matrix matches Table 1 in \cite{DeCarlo_{2011}}, the latter matches Table 7 in \cite{DeCarlo_{2012}}; see original papers for item details.

Based on each dataset' s Q-matrix and fraction subtraction expertise, we construct attribute interaction matrices as in formulas (23) and (24):

$$\mathbf{Q}_1^{\#} \in \mathbb{R}^{17\times 8}, \quad \mathbf{Q}_2^{\#} \in \mathbb{R}^{8\times 5}$$

where $\mathbf{Q}_1^{\#}$ is FRAC' s interaction matrix, selecting 17 typical interaction types from $2^8 - 9 = 247$ possible interactions among 8 attributes. $\mathbf{Q}_2^{\#}$ is Sub FRAC' s interaction matrix, selecting 8 typical types from $2^5 - 6 = 26$ potential interactions. For example, the first row of $\mathbf{Q}_2^{\#}$, $\mathbf{q}_{21}^{\#} = (1, 1, 0, 0, 0)$, indicates an interaction between attributes a and b in Sub FRAC—specifically, basic fraction subtraction must be completed before simplifying results to lowest terms. Other rows represent similar interactions.

Bi-QNN pre-training data is generated identically to the simulation study: based on each dataset' s Q-matrix, we generate pre-training simulated datasets using GDINA and DINA models, with 800 and 1600 high- and low-quality samples respectively. Since original empirical datasets lack expert-diagnosed attribute mastery patterns, we follow \cite{Wang_{{{et}}{{al}}}{2023}} by using GDINA and DINA predictions on complete datasets as ground truth, then sample training and test sets with $N = (50, 100, 200, 300, 500)$. Pre-training and experimental repetition counts match the simulation study. Bi-QNN uses identical hyperparameters across both datasets: 200 iterations, batch size 128, and learning rate 0.01.

6.2 Results

6.2.1 Subset Results Table 4 shows attribute and pattern classification accuracy on Sub FRAC across sample sizes. As in simulations, sample sizes range from 50 to 500. We also include PMR(K-1) metric, allowing one attribute classification error. Focusing on AMR performance, Bi-QNN outperforms all models across all subsamples. ANN ranks second at larger sample sizes ($N \geq 300$), followed by parametric models (DINA, GDINA), with nonparametric models (NPC, GNPC) performing relatively poorly.

Regarding sample size trends, all models show increasing AMR scores with sample size, similar to simulations. Among nonparametric models, NPC performs relatively steadily, while GNPC shows slight declines after $N > 100$, possibly due to its simultaneous consideration of connection and non-connection mechanisms. For PMR and PMR(K-1), scores are higher for PMR(K-1) due to its tolerance for one error. Trends mirror AMR results, with Bi-QNN maintaining overall advantages across Sub FRAC subsamples. At sample size 50, DINA slightly outperforms Bi-QNN in PMR.

6.2.2 Full Dataset Results Table 5 mirrors Table 4' s structure, showing results on the full FRAC dataset. Overall trends are similar to Sub FRAC, with Bi-QNN maintaining comprehensive leadership, followed by ANN. All models perform worse on FRAC than Sub FRAC. Notably, Bi-QNN' s AMR scores show little difference between FRAC and Sub FRAC, while PMR and PMR(K-1) scores decrease slightly less than other models.

Neural network methods (Bi-QNN, ANN) show larger performance advantages over parametric (DINA, GDINA) and nonparametric (NPC, GNPC) methods on FRAC than Sub FRAC, indicating greater advantages on more complex data. Particularly at sample size 50, Bi-QNN shows clear leads across AMR, PMR, and PMR(K-1) on FRAC, more pronounced than on Sub FRAC.

7. Discussion

7.1 Key Findings

7.1.1 Bi-QNN Model Design Bi-QNN' s design originates from the GDINA model, incorporating two computation streams for main and interaction effects, constrained by the Q-matrix and interactive Q-matrix respectively. This provides semantic meaning and interpretability to the network structure, similar to Dropout methods that enhance robustness \cite{Srivastava_{{{et}}*{{al}}}{2014}}. Bi-QNN' s width is also constrained by these matrices, enabling automatic adaptation to any dataset and freeing users from considering network depth and width—greatly improving usability and generalizability. This represents the key difference from fixed-structure neural network cognitive diagnosis methods \cite{Cui_{{{et}}*{{al}}}{2016}, Chen_{{{Yan}}_{{2021}}}}.

This design reduces Bi-QNN' s sensitivity to attribute count, addressing limitations of traditional parametric and nonparametric models that are highly sensitive to attribute count and struggle with reliability on high-attribute data \cite{Sen_{{{Cohen}}*{{2021}}}}. *Evidence appears in our experiments: using SD2 as baseline, all models perform better on SD1 and SD3, with SD1 showing greater improvement. This indicates that reducing attribute count improves performance more than increasing item count, confirming attribute count' s greater impact—consistent with past research \cite{Najera{{{et}}*{{al}}}{2021}}.* Bi-QNN performs better on both SD2 and SD3, especially showing 10% AMR and 15-20% PMR leads on low-quality SD2 and SD3 datasets. Empirical results also show smaller performance variations on FRAC, confirming Bi-QNN can mitigate attribute count effects and demonstrate stronger robustness on low-quality data.

7.1.2 Bi-QNN Training Method Neural network applications to cognitive diagnosis face difficulties in ensuring correct latent variable labels. Since most tests are one-time with low item reuse, attribute annotation is extremely challenging, leading to scarcity of datasets with labeled response patterns

and attribute mastery patterns, making neural network training difficult. \cite{Cui_{{{et}}{{al}}}{2016}} used ideal response and mastery patterns, while \cite{Xue_{{{Bradshaw}}{{2021}}}} *used DINA-based semi-supervised co-training. However, ideal patterns for a given Q-matrix are very limited \cite{de{{{la}}{{Torre}}}{2009}}*, especially ideal response patterns that rarely cover real examinee response patterns. Moreover, actual attribute mastery pattern categories are typically far fewer than ideal patterns, limiting generalization of ideal data-trained models. While DINA-based semi-supervised methods can use non-expert labeled data, DINA cannot express attribute interactions and is sensitive to initial labels and noise, limiting performance improvements.

This paper employs transfer learning with pre-training and fine-tuning. First, Bi-QNN is pre-trained on large simulated datasets, then fine-tuned on actual small datasets, enabling training with limited samples (e.g., $N = 100$). This not only maintains good performance across sample sizes but also enhances generalizability—crucial for educational testing practice.

Evidence from experiments shows Bi-QNN' s AMR and PMR scores vary more smoothly across sample sizes than other models, maintaining good performance even with small samples, especially on low-quality data. On low-quality SD2 and SD3, Bi-QNN leads in PMR while maintaining the most stable scores (standard deviations of 0.016 and 0.019). At sample size 50 on low-quality SD1, SD2, and SD3, Bi-QNN' s leads in AMR and PMR are most pronounced. On real datasets Sub FRAC and FRAC with narrower quality ranges, Bi-QNN leads across nearly all sample sizes.

7.1.3 Limitations and Future Directions Despite strong performance, Bi-QNN has limitations. While relatively robust to attribute count and sample size, it remains somewhat sensitive to item quality, though less than other models—performance degradation is about half that of parametric/nonparametric models, remaining around 0.05. When item quality is high but sample size very small ($N \leq 50$), neural network parameter estimation remains challenging, where nonparametric models retain advantages, especially with fewer attributes. For example, on high-quality SD1 at $N = 50$, GNPC leads all models with AMR=0.966 and PMR=0.907.

Additionally, while transfer learning reduces annotation costs, it adds computational overhead. The pre-training phase consumes more resources due to large simulated dataset generation and training. In our experiments, pre-training averages 20 seconds on CPU and 30 seconds on GPU; fine-tuning takes 1-12 seconds on CPU and 1-30 seconds on GPU depending on sample size; prediction time is under 1 second. Other models require only 1-5 seconds total. Detailed timing is in Appendix 1.

Furthermore, simulation data is primarily generated under non-compensatory model assumptions, so Bi-QNN' s performance on compensatory model data

requires further validation. Since simulated attributes lack concrete meaning, we did not deeply explore how interaction matrix settings affect performance—an important future research focus. This study focuses on dichotomous models, while polytomous models have richer real-world applications requiring further deep learning exploration. Current methods only use response outcomes, not response processes, though neural networks excel at multimodal learning. Future work could incorporate response process data to reduce sensitivity to guessing and slip \cite{Tian_{{{et}}{{al}}}{2023}}. While Bi-QNN implements attribute interactions via Q-matrix and interactive Q-matrix constraints, future work could incorporate domain-specific knowledge graphs to better represent knowledge relationships and improve performance.

7.2 Conclusions

This study proposes Bi-QNN, a Q-matrix constrained neural network cognitive diagnosis model trained via transfer learning. Findings indicate:

1. Bi-QNN effectively expresses item-attribute relationships with automatically adaptive network structure. Users only need to provide Q-matrix and attribute interaction matrix, offering good usability and generalizability. Experiments show Bi-QNN can mitigate performance degradation from increasing attribute count, maintaining good performance within a certain range.

2. Transfer learning-based training effectively addresses insufficient sample sizes in cognitive diagnosis scenarios. Learning item-attribute relationships from large simulated data and transferring to specific diagnostic datasets enables robust performance across sample sizes, particularly excelling with low-quality data and relatively more attributes.

3. Model selection recommendations: For very few attributes ($K < 5$) and small samples ($N < 50$), nonparametric models show advantages. However, Bi-QNN demonstrates better overall performance in most cases, especially on empirical data, making it a versatile model for various conditions.

4. Application scenarios: Given its usability, generality, and attribute count tolerance, Bi-QNN is ideal for psychological assessment tools developed by experts—once trained, models can be reused repeatedly. In education, Bi-QNN serves well as a universal assessment model in meta-learning systems, especially those with knowledge graph support where models can auto-train via test assembly algorithms. In classroom settings with relatively more attributes ($K > 5$) and sample sizes above 50, Bi-QNN remains an excellent comprehensive choice.

References

References are preserved exactly as in the original text

Note: Figure translations are in progress. See original paper for figures.