International Journal for Housing Science and Its Applications
Publish September 16, 2025. Volume 47, Issue 1 Pages 1080-1093

https://doi.org/10.70517/ijhsa47193

The Application of Composition Technology Theory in College Music Teaching Based on Edge Computing under Digital Platform

Hua Bai[1,*]
[1] School of Arts, Henan University of Economics and Law, Zhengzhou, Henan, 450046, China
Corresponding authors: (e-mail: 13903711261@163.com).

Abstract Integrating smart technology into the teaching of composition technology discipline can bring more innovative opportunities for the development of music education. In this paper, for the teaching of the theory course of composition technology, edge computing technology is combined with music education to propose a smart music teaching platform, and a knowledge tracking and exercise recommendation strategy is designed for this purpose. The strategy constructs a knowledge tracking model based on the self-attention mechanism and hypergraph module, on the basis of which the POMDP process is used to build exercise recommendation tasks to achieve personalized music exercise resource recommendation. Experiments show that the designed knowledge tracking model is more than 3% more accurate than the DKT method, and the mean value of each index of the exercise recommendation model is above 0.8, which is more targeted, novel and diversified. After the application of this teaching platform, students' attitudes and willingness towards the course, interaction and platform have increased significantly, by 25.25%, 30.53% and 37.68% respectively. Meanwhile, more than 60% of the students affirmed its enhancing effect on teacher-student interaction, exercise recommendation, course interest and learning effect. The proposed teaching platform effectively promotes teacher-student interaction teaching, provides personalized aids for music course learning, and can help promote the development of music teaching quality in colleges and universities.

Index Terms edge computing, self-attention, knowledge tracking model, exercise recommendation model, music teaching

I. Introduction

An excellent musical work centers around a certain theme, comprehensively utilizes the laws and techniques of music composition, and presents a complete musical story through the organization of musical language. In the whole process of music creation, both vocal music and instrumental music have to achieve the purpose of expressing philosophical thoughts and emotions through melody, rhythm, and comprehensive sound effects, and how to realize the perfect combination of melody, rhythm, sound effects and other elements is the problem to be solved by the theory of composition technology, which is indispensable to the theory of composition technology [1]-[3]. The importance of compositional technology theory in college music teaching can be summarized in two aspects. On the one hand, compositional technology theory can further improve the content of music teaching through its systematic and comprehensive knowledge of music technology theory, so that music teaching not only emphasizes on the teaching of music history and theory, but also emphasizes on the teaching of music sense, meter and compositional technology [4]-[6]. On the other hand, although the theory of composition technique is a music theory system, many of its contents are inclined to practical operation, such as the conception of polyphonic music weave [7]. These contents, on the basis of theoretical exposition, can guide students to master the use of the main key and polyphony, and learn to deal with the musical melody as well as the arrangement of the musical language, so as to comprehensively consolidate students' musical skills [8]. In addition, in all the music-related courses, the theory of composition technology is not only highly practical in content, but also includes the practice of music creation and composition technology of many genres from traditional to modern, so that students can come into contact with different music styles through appreciation, practice and other ways, so that students can learn from a variety of music styles in the process of micro-comparison learning, and can make up for the students' musical practice caused by the teaching content being too biased in favor of the theoretical. It can make up for the problem that students' practical ability is weak due to the fact that the teaching content emphasizes too much on theory [9]-[11]. However, in the process of teaching composition, the traditional indoctrination teaching method is still used, which leads to the low interest of students in learning, and is difficult to meet the current stage of society's requirements for music creation talents, resulting in poor comprehensive ability of students, which is not conducive to the development of the music industry [12], [13]. In addition, composition is the core link of music creation, and composition technology, as an important part of supporting the practice of composition, has a crucial impact on the quality and creative efficiency of composition [14]. However, there exists a unilateral tilt of composition technology and composition art in composition creation and teaching, resulting in a serious sense of fragmentation on the finished music, which is not conducive to artistic expression [15], [16].

With the advancement of digitization in various industries, music education and music creation have undergone a whole new transformation. From artificial experience-oriented to intelligent technology for music evaluation and guidance, from musical instrument digital interface technology to intelligent music generation form of music creation, all of them are marking the intelligent technology basis to become an indispensable tool in the field of music education and creation [17]-[19]. The digitization process has been accompanied by the rapid development of various technologies such as the Internet of Things, 5G communication, and cloud computing, which provides a platform for the development of edge computing. Edge computing is an emerging computing paradigm that integrates the core capabilities of network, computing, storage, and applications on an open platform, which moves computing resources from data centers to edge devices close to the data source in order to process data faster and provide a better user experience, presenting characteristics such as discrete, real-time, data-aware, and network edge [20], [21]. Edge computing and cloud computing synergize with each other to help digital transformation in various industries. It provides intelligent interconnected services in close proximity to satisfy the critical needs of the industry for real-time business, business intelligence, data aggregation and interoperability, security and privacy protection during the digital transformation process [22]-[24]. This provides the possibility of real-time, interactive, flexible, dynamic adjustment, and personalized services for music teaching and creation.

In this paper, we utilize edge computing technology to build a smart music teaching platform consisting of five parts, and explore its application teaching in the theory teaching of composition technology. Based on this, a hypergraph and self-attention mechanism are used to model students' historical interaction sequences, and the construction of a hypergraph self-attention knowledge tracking model oriented to knowledge states is realized by considering historically relevant interactions to predict students' interactions in the next exercise. Using this knowledge tracking model as a student simulator to complete the interaction with the environment, the exercise recommendation process is modeled as a POMDP process, and the exercise recommendation strategy is optimized by the REINFORCE algorithm based on the strategy gradient to obtain the knowledge tracking and exercise recommendation method based on music teaching. Three datasets are selected to carry out experiments to test the knowledge tracking performance of this paper's method as well as the accuracy, novelty and diversity of exercise recommendation through the comparative analysis of models. Finally, the teaching experiment of the smart music teaching platform is designed, and its application effect in the teaching of the theory course of composition technology is evaluated multidimensionally by means of a questionnaire survey.

II. Smart Music Teaching Platform Based on Edge Computing

With the continuous development of the Internet and 5G technology, the era of smart education has arrived. The great achievements of edge computing technology in the fields of smart city, smart transportation, smart industry, etc. provide favorable reference for the design of smart teaching environment. Relying on the existing resources of the intelligent environment, the intelligent music teaching platform based on edge computing is constructed as a means of realizing the intelligent teaching of the theory course of composition technology, making education more personalized and management more intelligent, and providing a powerful technical support for innovative teaching practice activities.

II. A. Overall framework

Based on the analysis of the advantages of edge computing technology and its application prospects in the intelligent teaching environment, this paper constructs the intelligent music teaching platform architecture based on edge computing technology as shown in Figure 1. By introducing edge nodes to pre-process the data collected by the terminal devices in real time using machine learning/artificial intelligence algorithms and make decisions quickly, it meets the requirements of intelligent, personalized, and real-time learning environment of the theory course of composition technology.

The intelligent music teaching platform based on edge computing technology is mainly composed of five layers: sensor layer, communication interface layer, edge computing layer, core network layer and cloud processing layer, as described below.

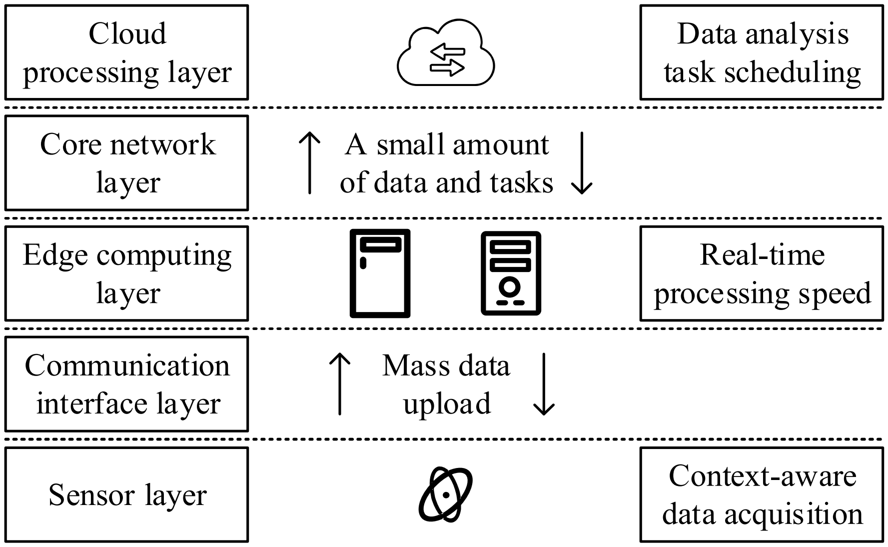

Figure 1: The intelligent music teaching platform architecture based on edge computing technology

II. A. 1) Sensor layer

The IoT devices in this layer can obtain multi-source heterogeneous data from the smart music classroom platform, such as environmental data (classroom temperature, humidity, CO2 concentration, etc.), identity data (student name, student number, face image, etc.), student gesture (head down, head up, emotion, etc.), device data (device location, operation status, malfunction message, etc.), and student physiological data from the wearable devices (electrical skin response emotional states, ECG signals, health detection data, etc.), etc.

II. A. 2) Communication interface layer

This layer is the one that connects to the edge nodes through various communication interface protocols (e.g., Wi-Fi, ZigBee, Bluetooth, RS232/485, etc.) and collects various heterogeneous and educational data from the sensors through synchronous or asynchronous modes.

II. A. 3) Edge computing layer

The edge computing layer is configured as a fixed edge node (SEN) or a mobile edge node (MEN) depending on the user requirements. Each edge node has its own storage and computing device for storing information about the data transmitted from the associated edge nodes, IoT sensing devices, and pre-processing and analyzing the collected data to speed up the decision-making process.

II. A. 4) Core network layer

The role of the core network layer is to transmit information to exchange data and act as a bridge between the edge computing layer and the cloud computing service layer. In general, the core network layer can be based on public telecommunication networks, the Internet, and industrial private communication networks (e.g., 4G/5G cellular mobile networks).

II. A. 5) Cloud processing layer

The cloud processing layer is the brain of the intelligent music teaching platform, which is mainly responsible for conducting in-depth analysis and mining of data processed by the edge modules to provide teachers and students with intelligent teaching support. The cloud processing center can use cloud computing technology and artificial intelligence algorithms to conduct in-depth analysis of the teaching data and discover the patterns and trends behind the data, so as to provide targeted teaching suggestions for teachers and personalized learning paths for students. In addition, the cloud processing layer contains different edge application service packages to provide support for updating and iterating edge nodes.

II. B. Pedagogical applications

With the continuous development of edge computing technology, the intelligent music teaching environment will be more intelligent, personalized and real-time. It is specifically manifested in the following aspects:

Intelligent recommendation of teaching resources. Edge computing technology can analyze students' learning needs and interests in real time, so as to recommend more appropriate and targeted music teaching resources for teachers and students, and improve the teaching effect.

Personalized learning path planning. Edge computing technology can customize personalized learning paths for students according to their learning progress and abilities, helping students complete their learning tasks more efficiently.

Intelligent teaching assistance. Edge computing technology can provide teachers with intelligent teaching aids, such as real-time generation of teaching feedback, automatic adjustment of teaching progress, etc., to help teachers improve the quality of music teaching.

Real-time student assessment and feedback. Edge computing technology can collect students' learning data in real time, provide teachers with accurate and comprehensive student performance data, help teachers better understand the students' learning status of the theory of composition technology course, and formulate targeted music teaching programs.

Online interactive teaching. Edge computing technology can reduce network latency, improve the real-time and interactivity of online teaching, and provide students with a more vivid and interesting learning experience.

III. Knowledge tracking model and exercise recommendation strategy

The research in this chapter is centered on a smart music teaching platform based on edge computing and proposes a knowledge tracking model and an exercise recommendation strategy. The model consists of two parts, the hypergraph self-attentive knowledge tracking model (HTNKT) for knowledge states and the exercise recommendation model.

III. A. Knowledge tracking model

Hypergraph Self-Attention Neural Network is a knowledge tracking model proposed in this paper, which combines hypergraph with self-attention. The hypergraph module not only groups exercises based on knowledge concepts, but also generates interaction embeddings for learners directly. After that, a gate recursion unit is used to obtain the students' knowledge states at different moments. For the self-attention module, this paper uses a converter to capture the long dependencies of the interaction sequences, assigning more attention weights to the previous exercises related to the current exercise. Finally, a gating mechanism is used to assess the student's mastery of a particular exercise in the current time period, thus predicting whether the learner will be able to correctly answer the provided exercise. In this paper, we improve the self-attentive knowledge tracking model by introducing a deeper attention mechanism module applied to knowledge tracking.

III. A. 1) Hypermap module

Hypergraph Self-Attention Knowledge Tracking constructs an interaction hypergraph structure for a music student's history doing sequence. The student's correct answer v_{i+} and incorrect answer v_{i-} to the i st exercise will be generated by hypergraph convolution to embed x_{i+} and x_i , respectively, and the interaction nodes belonging to the same knowledge concept construct the dependencies between interactions based on the hyperedges of the knowledge concepts:

$$h(v,e)=\begin{cases}1, & \text{if } v\in e\\ 0, & \text{if } v\notin e\end{cases} \tag{1}$$

$h(v,e)$ denotes the matrix representation of the generated hypergraph.

The hypergraph module of hypergraph self-attentive knowledge tracking extracts the features of the hyperedge using hypergraph convolution based on the features embedded in the learner's historical interactions and the hypergraph adjacency matrix. Then, the features of the hyperedge are subjected to an aggregation operation thus completing the update of the interaction embedded features:

$$Y=D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}}X\theta \tag{2}$$

$$D_v=\sum_{e\in E}\omega(e)h(v,e) \tag{3}$$

$$D_v=\sum_{v\in V}h(v,e) \tag{4}$$

where, H is the feature matrix of the hyperedge, D_v is the degree of the node and D_e is the degree of the hyperedge. ω is the weight matrix of the hyperedge initialized with all values of 1, and θ is the learnable parameters.

III. A. 2) Self-attention module

The self-attention module of the hypergraph self-attention knowledge tracking model is a converter-based encoder-decoder structure. Residual linking and layer normalization are applied at each layer of the self-attention module. First the learner's historical interactions are embedded as the encoder's attentional keys, values and queries. The attentional weights between interactions are computed and output, and then this information is returned to the decoder:

$$\begin{cases} M = LayerNorm(SkipConct(MHA(Q,K,V))) \\ O = LayerNorm(SkipConct(FFN(M))) \end{cases} \tag{5}$$

where M denotes the output of the encoder's self-attention layer, O denotes the output of the encoder, FFN is two superimposed linear layers, and $SkipConct$ is the residual connection. The first layer of the decoder uses multi-head attention to determine the attentional weights between the exercise embeddings, while the second layer computes the connections between the interactive activities and the exercises:

$$\begin{cases} M_1 = LayerNorm(SkipConct(MHA(Q,K,V))) \\ M_2 = LayerNorm(SkipConct(MHA(M_1,O,O))) \\ Dec = LayerNorm(SkipConct(FFN(M_2))) \end{cases} \tag{6}$$

where M_1 represents the output of the first self-attention layer of the decoder, M_2 represents the output of the second self-attention layer of the decoder, and Dec represents the final output of the decoder.

In conclusion the self-attention module of hypergraph self-attention knowledge tracking can be described as:

$$\begin{cases} O = Encoder(I) \\ Dec = Decoder(E,O) \end{cases} \tag{7}$$

The formula for the hypergraph self-attention knowledge tracking model of multiple attention is:

$$MHA = Concat(Soft\max(Mask(\frac{Q_1K_1^T}{2}))V_1, \cdots, Soft\max(Mask(\frac{Q_iK_i^T}{2}))V_i)W^O \tag{8}$$

where the product of Q_i and k_i represents the attentional weight of v_i, which is then divided by the square root of d, i.e., the dimensions of q and k, for dimensionality reduction. A mask is applied and the attentional weights activated by the softmax function are multiplied with v to obtain the output of the i th attention head. Finally, the outputs of all the attention heads are concatenated to form the output of the multi-head attention. The output of the multi-head attention is linear, and two layers of linear transformations with ReLU activation need to be added to enhance the representation of the hypergraph self-attention knowledge tracking. Then:

$$FFN(X) = \mathrm{Re}LU(xW_1 + b_1)W_2 + b_2 \tag{9}$$

where x denotes the output of the multi-head attention. W_1, W_2, W_1 and b_2 are the shared offset and weight matrices.

III. A. 3) GRUs and gating mechanisms

After initializing the learner's interaction hypergraph, the learner's interaction sequence corresponds to a path in the hypergraph, and in this paper, we use a cyclic gating unit (GRU) to further update the learner's knowledge state.

At each time step t, the reset gate r_t of the cyclic gating unit simulates the process of forgetting by the learner to decide which past information to forget, and the update gate u_r of the cyclic gating unit simulates the process of remembering by the learner to decide how much new information will be merged into the current knowledge state h_t. The new information to be added to h_t is represented by memory unit C_t. The current knowledge state is calculated by combining the knowledge state h_{t-1} of the previous time step with the storage unit C_t. Then:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \tag{10}$$

$$u_t = \sigma(W_u[h_{t-1}, x_t] + b_u) \tag{11}$$

$$c_t = \tanh(W_c[r_t \odot h_{t-1}, x_t] + b_c) \tag{12}$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot C_t \tag{13}$$

where the weight matrices of the reset gate, update gate and storage cell are denoted by W_r, W_u and W_c, respectively. b_r, b_u and b_e are the learnable bias terms of the reset gate, update gate and storage cell, respectively. $\odot$ shows the multiplication of the corresponding position elements between the matrices, and $\sigma()$ and $\tanh()$ are nonlinear activation functions.

The extent to which students understand various concepts depends on their knowledge state. When given Exercise $e_i \in E$, this paper relates the current knowledge state of the learner obtained from the hypergraph module to the embedding of the hypergraph nodes in order to represent the student's knowledge state s_i^h for Exercise e_i. The output of the selfattention module is utilized to obtain the learner's knowledge state s_i^1 for Exercise e_i. The hypergraph self-attentive knowledge tracking model uses a gating mechanism to integrate the knowledge states of the two modules in order to combine the respective strengths of the two modules. Then:

$$s_i^e = gate \odot s_i^h + (1 - gate) \odot s_i^t \tag{14}$$

The basic idea of the gating mechanism is described as follows:

$$gate = \sigma((W_{g1} s_i^h + b_{g1}) + (W_{g2} s_i^h + b_{g2})) \tag{15}$$

where the learnable weights of the hypergraph and attention modules are denoted by W_{g1} and W_{g2}, and the learnable offsets are denoted by b_{g1} and b_{g2}, respectively. To balance the importance of the hypergraph and self-attention modules a gating mechanism is added before the prediction layer. The predicted value $\hat{r}_i$ of the learner's answer to Exercise $e_i \in E$ is obtained from the knowledge state s_i^e by a linear layer using a sigmoid activation function:

$$\hat{r}_i = Sigmoid(s_i^e w_o + b_o) \tag{16}$$

III. B. Exercise recommendation model

In this paper, we use the POMDP process to model the exercise recommendation task and optimize the recommendation strategy using deep reinforcement learning algorithms. In it, the hypergraph self-attentive knowledge tracking model (HTNKT) model, which was proposed in the previous section to be oriented to the knowledge state, is used as a student simulator to help complete the interaction between the intelligences and the environment in reinforcement learning.

III. B. 1) Overall structure

The optimization process of the exercise recommendation model implemented using deep reinforcement learning in this paper is shown in Figure 2. In the personalized exercise recommendation task: the intelligent body is the recommendation algorithm with the function of pushing exercises and receiving feedback, the environment is the information of students and exercises, the state is the students' mastery level of knowledge, the action is the next practice behavior that the students will carry out, and the reward is the degree of improvement of the students' knowledge after doing the exercises.

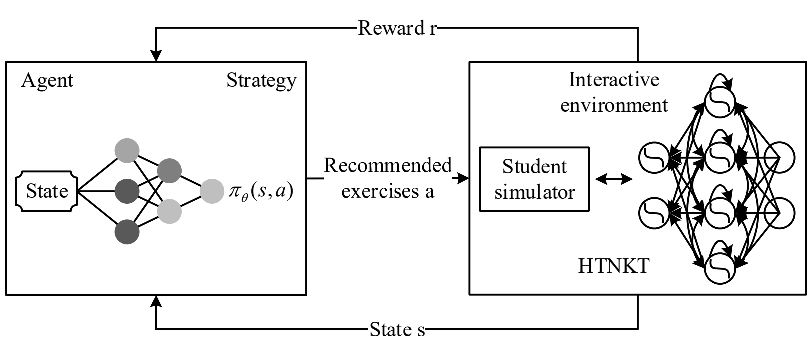

Figure 2: Process of exercise recommendation model optimization

III. B. 2) Decision-making process

Intelligence in the exercise recommendation task is defined as a decision-making entity, which is mainly represented by invisible computation of the parameters of each layer in the neural network of deep reinforcement learning. The environment is defined as a dynamic interaction scenario between the student and the recommendation model, which includes information about the student's knowledge level and practicable exercises, which is replaced in this paper by the knowledge tracking model constructed in the previous section.

There are different probabilities of choosing different actions from the initial state, using $\pi_\theta(a \mid s)$ to denote the probability of choosing action a in state s. In this paper, a neural network is used to fit the strategy, and θ denotes the parameters of the neural network. After selecting an action due to the randomness of the environment, the state transfer probability $p(s' \mid s, a)$ is used to denote the probability value of state transfer to the next state in this case. The trajectory of a round of continuous interaction between the intelligence and the environment until it is aborted can be expressed as follows: $\tau = \{s_0, a_0, s_1, a_1, \ldots s_T, a_T\}$. Since the state transfer probability is unknown and the student's response situation $o_t \in O$ can be observed, the effect of different observation situations on the state needs to be considered, and the trajectory can be transformed into $\tau = \{s_0, a_0, o_1, s_1, a_1, o_2, \ldots s_{T-1}, a_{T-1}, o_T\}$. The probability of the trajectory occurring in the case where the decision-making behavior is π is shown in Eq. (17):

$$p_\theta(\tau) = p(s_0)\prod_{t=0}^{T-1} p(s_{t+1} \mid s_t, a_t)\, p(o_{t+1} \mid s_{t+1}, a_t)\, \pi(a_t \mid o_t) \tag{17}$$

III. B. 3) Reward settings

Each decision-making action of the intelligent body will generate a single-step reward, which in this paper is defined as shown in Equation (18) for the single-step reward r_t obtained by performing action a_t in state s_t:

$$r_t = \frac{1}{K}\sum_{i=1}^{K} w_i P_{t+1}(q_i) \tag{18}$$

where K indicates the number of exercises under a certain knowledge point, $P_{t+1}(q_i)$ indicates the probability of correctly answering q the question when the learner's state is transferred to s_{t+1} after practicing the question a_t. The difficulty coefficient of the question is determined by the probability of correctly answering the question, i.e., the lower the rate of correctly answering the question, the higher the difficulty coefficient of the exercise, and the w_i weighting value is determined by the difficulty coefficient of the question. Thus the reward r_t is the weighted average of the probability of correctly answering each question under a given knowledge point in the current state of hidden knowledge.

The total reward for a round is the sum of the individual rewards for all actions and can be denoted as $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$. Where $\gamma \in (0,1]$ is the reward discount factor, which is used to balance current and future rewards and helps optimize the long term decision making of the intelligence. Overall the decision trajectory rewards under a single round are weighted by the probability of their occurrence, and the expected reward of Strategy $\pi_\theta(a \mid s)$ is the sum of all trajectory rewards computed. However, the state transfer probability of the environment is unknown and it is not possible to exhaust all possible trajectories that can be generated. In this paper, we approximate the expected reward by using Monte Carlo sampling method, and the more samples the more accurate the approximation. The final expected reward $\overline{R}_\theta$ is shown in Equation (19) and $\overline{R}_\theta$ is the objective function used to optimize the neural network:

$$\overline{R}_\theta = \sum_{\tau} R(\tau) p_\theta(\tau) \approx \frac{1}{N}\sum_{n=1}^{N} R(\tau) \tag{19}$$

III. B. 4) Strategy Optimization

This section describes the process of optimizing the model parameters using the REINFORCE algorithm, which is a gradient-based policy optimization algorithm implemented through the Gym library of reinforcement learning algorithms. The optimization objective of the strategy gradient algorithm is to maximize the expected return $\overline{R}_\theta$, and the strategy parameters are updated by solving for the gradient of the objective function with respect to the strategy parameters θ. The formula for calculating the policy gradient is shown in (20), where N represents the number of trajectories, and for each trajectory the corresponding cumulative reward $R(\tau)$ and its corresponding gradient can be calculated:

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R_t(\tau^n)\nabla \log \pi_\theta(a_t^n \mid s_t^n) \tag{20}$$

In the previous subsection it was shown that the cumulative reward $R(\tau)$ of the trajectory is a summation of the reward of each step of the action, and this section gives a more detailed treatment on the code implementation of the REINFORCE algorithm. Specifically, the reward value $[r_1, r_2, r_3, ..., r_T]$ for each action step in the whole round can be obtained based on the trajectory, and the cumulative reward value $R_t(\tau)$ after performing the action a_t at time t can be set as shown in Equation (21):

$$R_t(\tau) = r_{t+1} + \gamma R_{t+1}(\tau) \tag{21}$$

$R_t(\tau)$ is determined by the reward generated by the action and the cumulative reward behind it, in the code implementation the sequence of rewards is traversed from back to front to determine the future rewards first, so that the cumulative reward behind each action can be obtained to optimize the decision making behavior.

The strategy gradient algorithm updates the parameters by a gradient ascent method, and the update formula is shown in (22), where α is the learning rate, which is used to control the step size of the parameter update. Based on the gradient its original function can be inverted as a loss function as shown in equation (23):

$$\theta \leftarrow \theta + \alpha \nabla \bar{R}_\theta \tag{22}$$

$$Loss = -R_t(\tau)\log \pi_\theta(a_t \mid s_t) \tag{23}$$

In this paper, the Adam optimizer is used to update the model parameters, while the objective of the optimization is to maximize the expected return, so the loss function is set to a negative value.

III. C. Experimental design

III. C. 1) Experimental data

(1) ASSISTment: This dataset is an educational dataset that is widely used in the field of educational data mining. It has been used to develop assessment intelligence machine learning models to assess students' cognitive and skill mastery status, and to provide students with personalized guidance to help them complete adaptive learning and improve learning efficiency. The dataset contains more than 100,000 student responses to math questions collected from online learning platforms, including question sequences, student information, and other data. The datasets used in this experiment are ASSISTment2009 and ASSISTment2015.

(2) Math dataset: this dataset is from Wisdom Learning Network provides an information environment and service for schools to provide personalized teaching of homework, exams, and curriculum education. The dataset contains information such as final exam questions from several high schools, and student answer records.

Since the dataset comes from some educational platforms and public datasets, some of the data have anomalies such as missing and duplicates. Therefore, it is necessary to deal with the anomalies of the data first before the experiment. The data in the dataset is divided into three parts: training set, validation set, and test set according to the ratio of 7:2:1.

III. C. 2) Knowledge tracking performance

In this experiment, a five-fold cross-experimental validation approach is used in order to measure the hypergraph self-attentive knowledge tracking model (HTNKT) oriented to the knowledge state, and to compare it with the existing classical knowledge tracking models, three metrics, accuracy (ACC), area under the ROC curve (AUC), and root mean square error (RMSE), are used as the performance measures.

The ACC metric, also called accuracy, indicates the percentage of correct predictions a model can make on a test dataset, and measures the performance of the classifier.AUC is defined as the area under the ROC curve and surrounded by the axes, and is often used as a metric to estimate the performance of the learner.The value of the area under the ROC curve ranges between 0.5 and 1. When the AUC value is equal to 0.5, the accuracy is the lowest and has no application value. When it is greater than 0.5, the closer it is to 1, the higher the detection performance of the model. Root Mean Square Error (RMSE), also called standard error, is often used in machine learning to measure the error between the predicted and true values.

In order to measure the performance of the HTNKT method proposed in this paper, the baseline method is chosen to compare the DKT model, DKVMN model. The results of knowledge tracking performance of different models are shown in Table 1. On different datasets, the proposed knowledge state-oriented hypergraph self-attentive knowledge tracking model improves the AUC values by 3.22%, 4.70%, and 3.45%, respectively, compared to the DKT method. Compared to the DKVMN method, it improves by 2.37%, 3.27%, and 2.84%, respectively. Similarly, both ACC and RMSE evaluation metrics are better than the other two classical models to get better knowledge tracking results.

Table 1: Knowledge tracking performance of different models

Index	Dataset	DKT	DKVMN	HTNKT
AUC	ASS09	79.21%	80.06%	82.43%
	ASS15	72.24%	73.67%	76.94%
	Math	73.90%	74.51%	77.35%
ACC	ASS09	70.01%	71.08%	74.83%
	ASS15	66.73%	68.84%	72.99%
	Math	72.05%	73.09%	76.62%
RMSE	ASS09	27.92%	26.29%	24.05%
	ASS15	28.34%	27.99%	24.86%
	Math	26.34%	24.25%	21.88%

III. C. 3) Recommended model evaluation indicators

To evaluate the effectiveness of the recommendation strategies proposed in the paper. A variety of exercise recommendation strategies are selected for comparison tests on each of the three datasets. And the following comparison tests are selected: User-CF, DKT and DKT-CF.

In the field of educational recommendation, the evaluation index of traditional recommendation is also generally referred to. The purpose of personalized exercise recommendation for music teaching is to personalize music exercise recommendation for users according to the learning characteristics of different users, the level of mastery of the state of theoretical knowledge of composition technology, and to help users to be able to improve in a targeted way. In this paper, accuracy, novelty and diversity are used to evaluate and compare the effects of different music exercise recommendation models. The novelty of the recommendation system is evaluated by calculating the similarity or distance between the recommendation results and the user's previous behavior. Diversity is used to measure the difference between recommended exercises, where different diversities can be defined using the similarity metric function, and in this paper, cosine similarity is used for similarity measurement.

III. C. 4) Analysis of the results of comparative experiments

In order to compare different exercise recommendation strategies, different methods are selected for comparison experiments in the above data sets, respectively. Comparison of accuracy, novelty and diversity of the three indicators on the list of recommended exercises, respectively, where the mean value Mean represents the mean value of the corresponding indicator on the recommended list, and Std.D represents the standard deviation of the corresponding indicator on the recommended list, and the smaller the value means that its performance is better.

The results of the accuracy comparison of the exercise recommendations are shown in Fig. 3. The hypergraph self-attentive knowledge tracking based knowledge state-oriented exercise recommendation method (HTNKT-CF) has a significant advantage over other baseline methods, with accuracy rates of 0.873, 0.858, and 0.819 for exercise recommendations on the ASS09, ASS15, and Math datasets, respectively.The DKT-CF recommendation method, because it uses the students' knowledge mastery state scoring matrix to do the collaborative filtering recommendation, has slightly higher accuracy and stability than the traditional User-CF and DKT methods. The HTNKT-CF method, on the other hand, achieves more stable results in exercise difficulty recommendation by combining the user knowledge mastery state in the exercise recommendation.

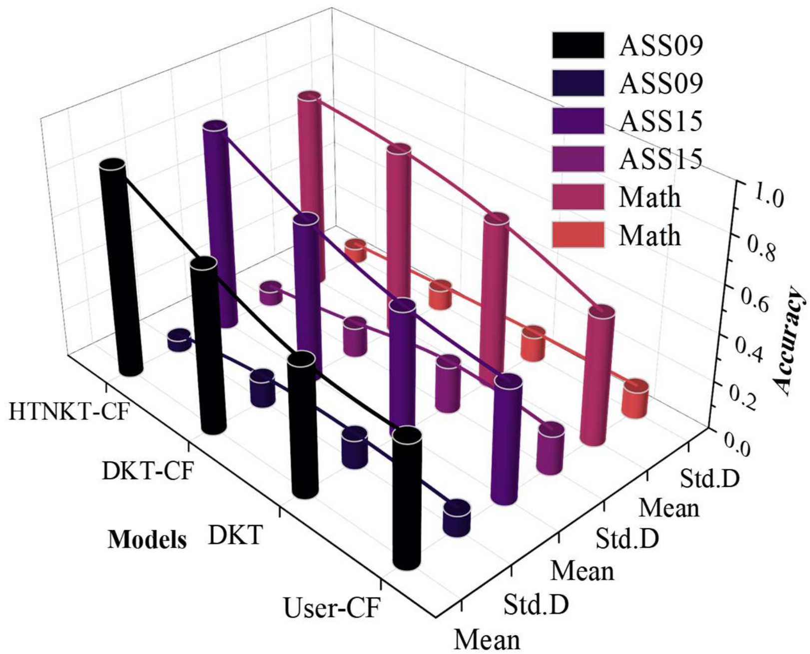

Figure 3: Comparison results of the recommended accuracy of exercise

The novelty comparison results of the exercise recommendations are shown in Figure 4, demonstrating the difference between the four different models on the novelty index of the exercises. The results of the novelty index of this paper's HTNKT-CF method on the three datasets are 0.835, 0.768 and 0.815, while the results of the novelty index of the other two methods are below 0.72. This paper's method improves the novelty of the method of recommending the exercises of the theory of compositional techniques in music teaching.

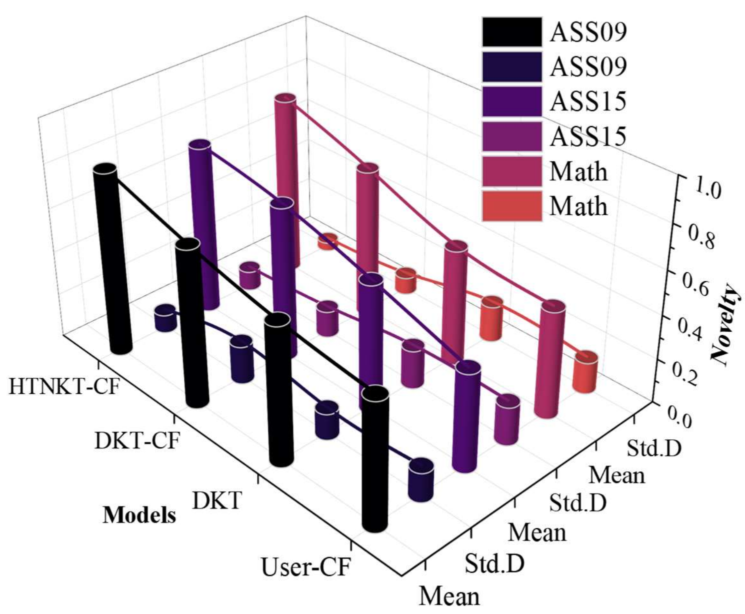

Figure 4: Comparison results of the recommended novelty of exercise

Figure 5 shows the differences in diversity metrics of different modeling methods. The diversity index of the HTNKT-CF method proposed in this paper is above 0.8 on all three datasets, with an average value of 0.825, while the average values of the diversity of the User-CF, DKT, and DKT-CF methods are 0.710, 0.753, and 0.792, which are lower than that of this paper's method. Overall, the recommendation model proposed in this paper can better improve the diversity of music exercises compared to other recommendation models.

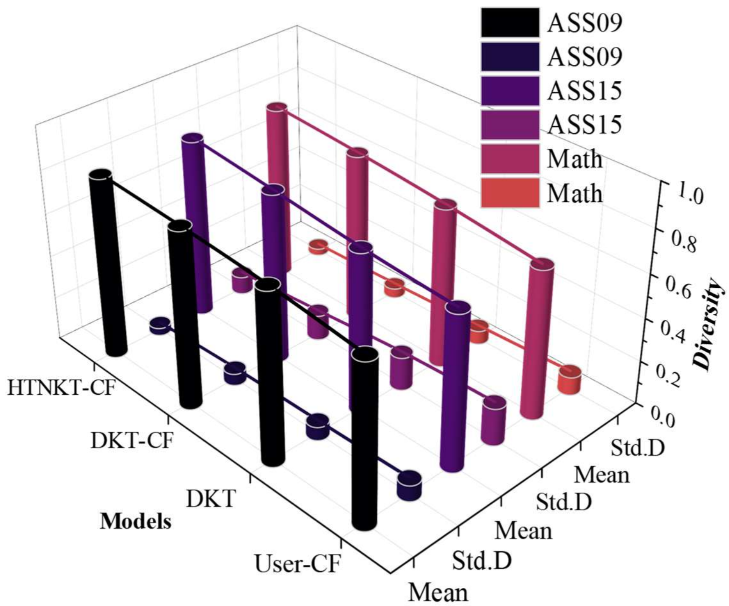

Figure 5: Different models differ in diversity indicators

IV. The application effect of the intelligent music teaching platform

This chapter conducts a semester-long classroom experiment in compositional technology theory to test the effectiveness of using an edge computing-based smart music teaching platform.

IV. A. Experimental design

The project deployed the Smart Music Teaching System between September and January 2024 on four classes of Theory of Composition Techniques at a conservatory. The experiment thus started in September and ended in December, lasting 4 months. A total of two teachers participated in the study, each teaching two classes with a total of 215 students.

Prior to the field deployment, the participating teachers were introduced to the features and UI interactions of the Smart Music Teaching Platform, and then their students were introduced to the system in the classroom, as well as to the Knowledge Tracking Nurturing Exercise recommendation model. After the introduction to the students, this paper conducted a pre-deployment survey via questionnaire with students in four classes to obtain initial measures of student opinions, including attitudes toward course learning, willingness to interact with teachers and students, and attitudes toward the teaching platform.

At the end of the 4-month classroom deployment, post-deployment surveys were administered to students and faculty. In the student survey, in addition to the same indicators as in the pre-deployment survey, four additional indicators were added: perception of changes in faculty-student interactions, effectiveness of exercise recommendations, course interest, and course learning effectiveness.

IV. B. Experimental results

IV. B. 1) Changes in attitudes and willingness

Teaching and learning attitudes, willingness of students to interact with teachers and students, and changes in attitudes toward the teaching platform were analyzed after students used the teaching platform. The results of students' attitude and willingness to change are shown in Figure 6. Students' attitudes towards course learning, willingness to teacher-student interaction and teaching platform attitude have significantly improved from 3.01, 2.85 and 2.76 points before the practice of the course “Theory of Composition Technology” to 3.77, 3.72 and 3.80 points, which is an increase of 25.25%, 30.53% and 37.68%, respectively. In terms of attitude towards course learning, the average value changed from slightly positive to more positive, and in terms of willingness to interact with teachers and students, the average value changed from slightly reluctant to more willing. There is a very significant shift in students' attitudes towards the teaching platform after using it. Before deploying the system, most of the students' attitudes were distributed from neutral to slightly opposed, and in general, they were slightly opposed to the deployment of the system. And after deployment, there was a significant shift in students' attitudes, which shifted to more favorable overall.

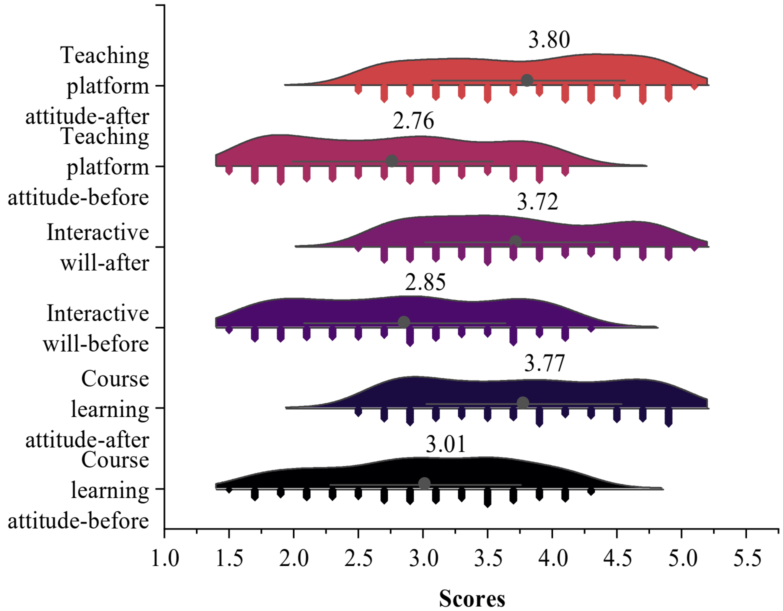

Figure 6: Changes in attitudes and willingness of students

IV. B. 2) Overall learning outcomes

The overall music course learning effect was analyzed in terms of students' perceptions of changes in teacher-student interactions, effectiveness of exercise recommendations, course interest, and evaluation of course learning effects. The overall music course learning effect is shown in Figure 7. 64.92% of the students perceived that the classroom interaction increased a little bit, and 11.43% of the students perceived that the teacher-student interaction increased a lot. At the same time, the number of students who thought that they had gained an increase in the evaluation of the effect of exercise recommendations, course interest and course learning effect were all above 60%, and the number of students who thought that they had gained a large amount of increase was 9.8% to 13.57%. It indicates that the use of the smart music teaching platform can promote the online interactive teaching between teachers and students in the course of Theory of Composition Techniques, the proposed knowledge tracking and exercise recommendation strategy has a better effect of music exercise recommendation, and the students gained the enhancement of learning interest and learning effect level.

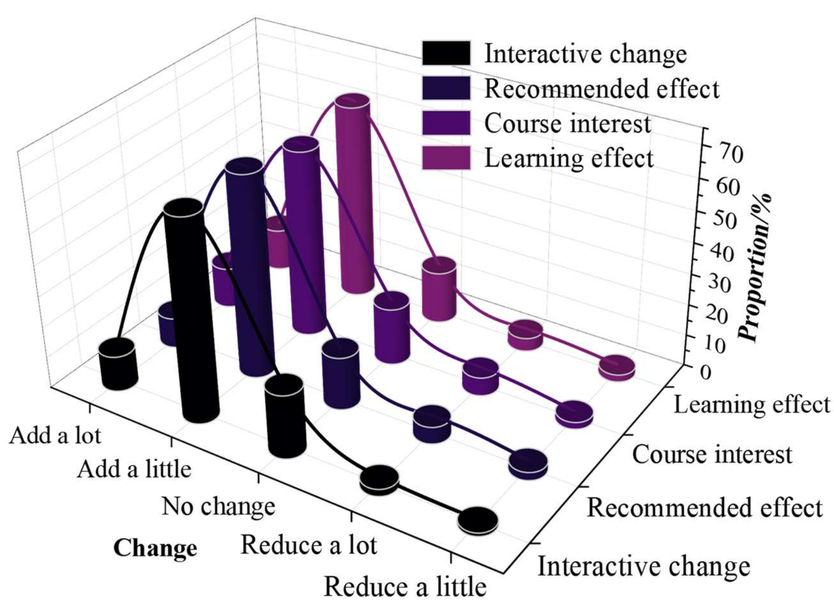

Figure 7: The learning effect of the whole music course

V. Conclusion

The study designs a smart music teaching platform based on edge computing, and constructs a knowledge state-oriented hypergraph self-attention knowledge tracking model and a music exercise recommendation model, and carries out experiments to explore the knowledge tracking and exercise recommendation performance of the method. Then the music teaching platform is used in the teaching of the theory course of composition technology, and its application effect is evaluated through questionnaire surveys.

(1) In different datasets, the knowledge state-oriented hypergraph self-attentive knowledge tracking model proposed in this paper outperforms the comparison method in all indicators, and the ACC value is more than 3% higher than that of the classical DKT method, which has better knowledge tracking performance. In terms of accuracy, novelty and diversity of exercise recommendations, the experimental average results of this paper's method are all greater than 0.8, which can provide students with more accurate, novel and diversified exercise practice.

(2) At the end of the teaching practice of the smart music teaching platform, students' attitudes toward course learning, willingness to interact with teachers and students, and attitudes toward the teaching platform increased by 25.25%, 30.53%, and 37.68%, respectively. And more than 60% of the students agree that the change of course teacher-student interaction, the effect of exercise recommendation, the interest in the course and the learning effect of the course get some improvement. The experiment proves that the music teaching platform based on edge computing can well assist the teaching of the theory course of composition technology, improve the online interaction between teachers and students and students' learning interest, and improve the quality of music teaching.

The application of edge computing and intelligent technology in intelligent music teaching is promising and will have a profound impact on the music education industry. The government, schools, enterprises and all sectors of the society should work together to promote the wide application of edge computing and intelligent technologies in the field of education, and make positive contributions to improving the quality of music education in colleges and universities.

Funding

Horizontal project"<Let music expel the haze, Embrace the beauty with Music> -- Music Therapy Intervention Project to intervene the negative emotions of special People"; Musical Psychological Counseling Studio.

References

[1] Knust, M. (2025). The Creation of Music: A Historical Overview of the Composition Process. In Pop Music Made in Småland: Music Production and Entrepreneurship in Sweden (pp. 13-28). Cham: Springer Nature Switzerland.

[2] Flieder, D. (2024). Towards a mathematical foundation for music theory and composition: a theory of structure. Journal of Mathematics and Music, 1-27.

[3] Iliya, D. B., & Achikeh, R. S. P. (2024). Trends in contemporary compositional techniques: An overview. AWKA JOURNAL OF RESEARCH IN MUSIC AND THE ARTS (AJRMA), 17.

[4] Pellegrino, K., Beavers, J. P., & Dill, S. (2019). Working with college students to improve their improvisation and composition skills: A self-study with music teacher educators and a music theorist. Journal of Music Teacher Education, 28(2), 28-42.

[5] Jing, Y. (2020). The Construction of Composition Technology Theory Curriculum System in Music Education of Normal University under the Background of "Harmony Centralization". Frontiers in Educational Research, 3(6).

[6] Eiksund, Ø. J., & Reistadbakk, E. (2020). Knowledge for the future music teacher: Authentic learning spaces for teaching songwriting and production using music technology. Music technology in education–channeling and challenging perspectives.

[7] Poe, E. A. (2021). The philosophy of composition. Graphic Arts Books.

[8] Tsougras, C. (2022). Applying the generative theory of tonal music to world music idioms: An analytical approach to the polyphonic singing of Epirus. Trends in World Music Analysis, 260-282.

[9] Remeš, D. (2019). New Sources and Old Methods: Reconstructing and Applying the Music-Theoretical Paratext of Johann Sebastian Bach's Compositional Pedagogy. Zeitschrift Der Gesellschaft Für Musiktheorie , 16(2), 51-94.

[10] Dean, R. T. (2017). Creating music: Composition. In The Routledge companion to music cognition (pp. 251-263). Routledge.

[11] Wen, Y. W., & Ting, C. K. (2022). Recent advances of computational intelligence techniques for composing music. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(2), 578-597.

[12] Junchaya, R. (2019). Teaching music composition. Creating form in time. Unigrafia.

[13] Hutchings, P. E., & McCormack, J. (2019). Adaptive music composition for games. IEEE Transactions on Games, 12(3), 270-280.

[14] Lörch, L., & Huovinen, E. (2025). Fostering composer voice in tertiary teaching of contemporary music composition. Journal of Research in Music Education, 73(1), 87-108.

[15] Vasil, M., Weiss, L., & Powell, B. (2019). Popular music pedagogies: An approach to teaching 21st-century skills. Journal of Music Teacher Education, 28(3), 85-95.

[16] Laato, S., Rauti, S., & Sutinen, E. (2020, July). The role of music in 21st century education-comparing programming and music composing. In 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT) (pp. 269-273). IEEE.

[17] Yang, W., Shen, L., Huang, C. F., Lee, J., & Zhao, X. (2024). Development status, frontier hotspots, and technical evaluations in the field of ai music composition since the 21st century: A systematic review. IEEE Access.

[18] Qiusi, M. (2022). Research on the improvement method of music education level under the background of AI technology. Mobile information systems, 2022(1), 7616619.

[19] Frid, E. (2019). Accessible digital musical instruments—a review of musical interfaces in inclusive music practice. Multimodal Technologies and Interaction, 3(3), 57.

[20] Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., ... & Das, S. K. (2022). Edge-computing-driven internet of things: A survey. ACM Computing Surveys, 55(8), 1-41.

[21] Nastic, S., Rausch, T., Scekic, O., Dustdar, S., Gusev, M., Koteska, B., ... & Prodan, R. (2017). A serverless real-time data analytics platform for edge computing. IEEE Internet Computing, 21(4), 64-71.

[22] Duan, Q., Wang, S., & Ansari, N. (2020). Convergence of networking and cloud/edge computing: Status, challenges, and opportunities. IEEE Network, 34(6), 148-155.

[23] George, A. S., George, A. H., & Baskar, T. (2023). Edge computing and the future of cloud computing: A survey of industry perspectives and predictions. Partners Universal International Research Journal, 2(2), 19-44.

[24] Qi, Q., & Tao, F. (2019). A smart manufacturing service system based on edge computing, fog computing, and cloud computing. IEEE access, 7, 86769-86777.