



Graph Deep Learning Model for Mapping Mineral Prospectivity

Renguang Zuo¹ · Ying Xu¹

Received: 13 May 2022 / Accepted: 13 July 2022 / Published online: 23 August 2022
© International Association for Mathematical Geosciences 2022

Abstract Mineral prospectivity mapping (MPM) aims to reduce the areas for searching of mineral deposits. Various statistical models that have been successfully adopted to delineate prospecting regions for a specific type of mineral deposit can be divided into pixel-wise, image- (or pixel-patch), and graph-based approaches. The pixel-wise models, which frequently integrate multiple prospecting information (or evidence layers) at a single pixel, do not adequately consider the spatial associations among neighboring pixels and may ignore the spatial patterns linked to mineralization or the spatial distribution characteristics of mineral deposits to some extent. Image-based models such as convolutional neural networks (CNNs) can extract local meaningful features and capture the spatial patterns of prospecting information in MPM because the input data of image-based models are images composed of regular pixels in the Euclidean space. However, CNNs also have limitations in MPM, such as the requirement for regular input data and non-rotationally invariant spatial features. Graphs that are typically composed of nodes and edges have a strong abstraction to capture the complex and nonlinear spatial relationships among multiple nodes and their edges. Prospecting information or evidence layers can be regarded as graphs in which pixels are connected by their adjacent pixels. In this study, graph deep learning algorithms, including graph convolutional networks and graph attention networks, were employed to produce mineral potential maps. A comparative study of graph deep learning algorithms with a CNN demonstrated the advantage of graph deep learning algorithms for MPM in terms of the cumulative areas versus the cumulative number of mineral deposits and the true/false prediction rate plot. These results suggest that the graph-based models, such as graph neutral networks, can effectively capture

✉ Renguang Zuo
zrguang@cug.edu.cn

¹ State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Wuhan 430074, China

mineralization information and the spatial interrelations between mineralization and prospecting information.

Keywords Mineral prospectivity mapping · Graph deep learning · Graph convolutional network · Graph attention network · Convolutional neural network

1 Introduction

The main aim of mineral prospectivity mapping (MPM) is to delineate prospecting areas to search for mineral deposits. There are many available models for MPM, which include knowledge-driven, data-driven, and hybrid models (Bonham-Carter 1994; Carranza 2008). In these models, weights of evidence (Agterberg 1989) and random forest (Rodriguez-Galiano et al. 2014) are popular because the former has an intuitive meaning for the weights of prospecting information (or evidence layers) and allows construction of input layers that have missing data (Porwal and Carranza 2015), and the latter is a kind of ensemble learning algorithm which has a strong ability to handle the complex and non-linear spatial associations between mineral deposits and geological features (Rodriguez-Galiano et al. 2014; Carranza and Laborte 2015; Zuo 2020). Most of these models can be regarded as pixel-wise approaches (Fig. 1a), which frequently integrate prospecting information from multiple sources at a single pixel in which the value is assigned either based on the spatial relationships between the locations of known mineral deposits and prospecting information or in terms of experts' knowledge on the formation of mineral deposits. However, when information available at a single pixel is fed into the pixel-wise approaches, they do not adequately consider the spatial associations among neighboring pixels and may ignore the spatial patterns linked to mineralization or the spatial distribution characteristics of mineral deposits to some extent (Talebi et al. 2022).

Recently, deep learning algorithms were employed in MPM because of their strong ability of feature extraction and integration (Zuo et al. 2022). These deep learning algorithms include deep autoencoders (Xiong and Zuo 2018), convolutional neural networks (CNNs) (Li et al. 2021b), long short-term memory networks (Wang and Zuo 2022), generative adversarial networks (Li et al. 2022), gated recurrent unit models (Yin et al. 2022), and variational autoencoders (Zuo et al. 2022). Among these models, CNNs are popular (e.g., Sun et al. 2020; Yang et al. 2021, 2022). A CNN can be regarded as an image-based (pixel-patch) model because the input data of CNNs are images which are composed of regular pixels in the Euclidean space (Fig. 1b). CNNs can exploit the local connectivity, shift-invariance, and compositionality of images (Bronstein et al. 2017) and, therefore, can efficiently identify local meaningful features (Wu et al. 2021, 2022) and capture the spatial pattern of prospecting information in the Euclidean domain.

Graphs, which are typically composed of nodes and edges, have a strong abstraction that can be adopted to encode arbitrary data types such as multidimensional data (Wu et al. 2022). An image can also be regarded as a graph where pixels are connected with their adjacent pixels and their connections which can be considered as edges. Compared with a graph, the node neighbors in an image are ordered and have a fixed

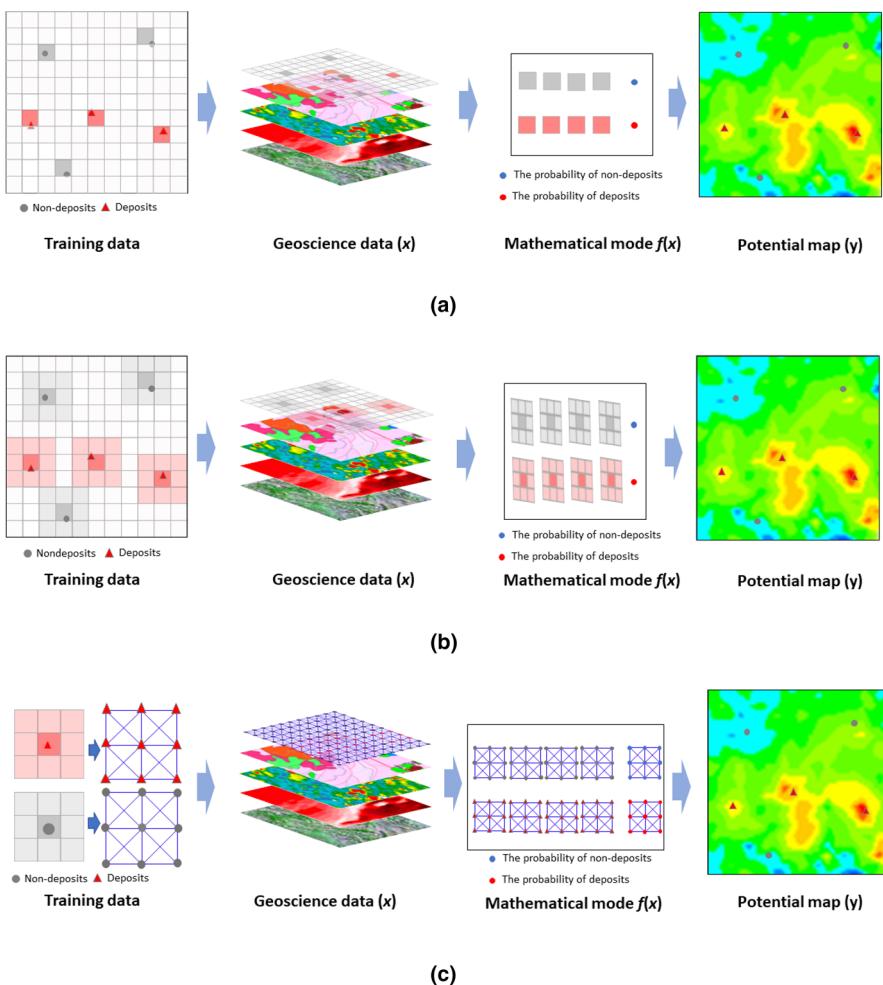


Fig. 1 Diagrams showing **a** pixel-wise, **b** image-based (pixel-patch), and **c** graph-based mineral prospectivity mapping

size, while the node neighbors in a graph are unordered and have a variable size (Wu et al. 2021; Guan et al. 2022). A graph has a strong ability to capture the complex and nonlinear spatial relationships among multiple nodes and their edges; therefore, it can effectively seize the mineralization information and the spatial interrelations between mineralization and prospecting information.

A graph neural network (GNN) is a type of graph-based model (Fig. 1c) that can capture hidden patterns of graph data (Gori et al. 2005). A graph convolution can be generalized from a two-dimensional convolution (Wu et al. 2021) in which each pixel in an image can be considered as a node where neighbors can be determined in terms of geological constraints. The convolution operation of GNN is similar to that in CNNs, which is an information aggregating and updating process. For updating

information operations, a two-dimensional convolution takes the weighted average of pixel values of a specific node along with its neighbors, while a graph convolution takes the weighted average value of the node features of a targeted node along with its neighbors. As a result, the information of the adjacent nodes can be learned and joined to the target node. CNNs frequently generate and train convolution features according to spatial positions, and may ignore image topological information. Taking an image with a size of 3 pixels \times 3 pixels as an example, the neighboring nodes of the center pixel are the eight surrounding pixels. All the pixels within this image have the same predictive probabilities in a CNN, while each pixel within this image may have a different probability in a GNN. There are three general types of predictive tasks in GNNs: graph-level, node-level, and edge-level. For a graph-level (Chen et al. 2022; Du et al. 2021; Fu et al. 2021), a node-level (Kipf and Welling 2017; Li et al. 2020a), and an edge-level task (Kipf and Welling 2017), a single property for a whole graph, some properties for each node in a graph, and the existence of links between nodes can be predicted, respectively. In this study, three different GNN models, namely, the graph convolutional network (GCN), graph attention network (GAT), and graph (multi-head) attention network (GAT*), were employed to produce a mineral potential map. A case study was implemented to demonstrate the advantage of the graph-based approaches compared with an image-based (i.e., CNN) model.

2 Methods

2.1 Graph Neural Network (GNN)

A GNN is a graphical approach that aggregates neighborhood node features to compute a new feature vector layer by layer. The intuitive idea of GNNs is that nodes in a graph denote objects or concepts and edges represent their relationships (Gori et al. 2005). The earliest GNN can be traced back to recursive neural networks (RNNs). Fransconi et al. (1998) and Sperduti and Starita (1997) used RNNs to process directed acyclic graphs. These pioneering studies inspired early research on GNNs. The concept of GNN was first proposed by Gori et al. (2005) and further developed by Scarselli et al. (2009) and Gallicchio and Micheli (2010). A GNN is a generalization of an RNN to some extent: (1) a GNN can process more general graphs, including directed, undirected, labeled, and cyclic graphs, and (2) a GNN can be applied both on graph- and node-focused problems (Gori et al. 2005).

Two GNN models, the graph convolutional network (GCN) and graph attention network (GAT), which are supervised learning algorithms, are introduced in this study. By using the back propagation algorithm, weight vector \vec{a} (a vector composed of multiple sets of attention coefficients) in GAT and weight matrix W in GCN and GAT are learned to minimize the difference between the predictive and real value. Graphs can be divided into directed and undirected graphs. The edges of a directed graph point from one node to another are irreversible, but they are invertible in an undirected graph. If two points are adjacent, they are joined by a pair of edges in an undirected graph (Wu et al. 2021).

A topology graph should be constructed to reflect node adjacency relations for GNNs. The topology graph determines the number of nodes that update node information, which is similar to convolutional kernels in CNNs. An undirected graph $G = (N, E)$ can be obtained by learning the spatial structures of point data.

$$N = \{\vec{n_1}, \vec{n_2}, \dots, \vec{n_m}\}, \vec{n_i} \in R^F, \quad (1)$$

$$E = \{E_{1,1}, E_{1,2}, \dots, E_{m,m}\}, E_{i,j} = \begin{cases} 1, & d_{i,j} \leq K \\ 0, & d_{i,j} > K \end{cases}, \quad (2)$$

where N and E represent the set of nodes and edges, respectively. If there is an edge between node i and node j , $E_{i,j} = 1$; otherwise, $E_{i,j} = 0$. Here, m represents the number of nodes in a graph G , $\vec{n_i}$ represents a node vector with several attributes, F denotes the number of features in each node, $d_{i,j}$ represents the distance between node i and node j , and K is the distance threshold; if $d_{i,j} \leq K$, node i and node j are connected.

2.2 Graph Convolutional Network (GCN)

Graph convolutions can be defined from a spectral perspective and a spatial domain. The spectral-based graph convolutions regard graphs as a signal and apply the eigenvalues and eigenvectors of the Laplace matrix of graphs to investigate graph properties. The spatial-based graph convolutions aggregate node representations from the neighborhoods. Kipf and Welling (2017) proposed a GCN to achieve efficient first-order neighbor convolution and aggregation through Chebyshev polynomials taking the first-order approximations. A multi-layer GCN with the following layer-wise propagation rule,

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \cdot H^{(l)} \cdot W^{(l)}\right). \quad (3)$$

Here, $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph with added self-connections. I_N is the identity matrix, $\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$, and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function $\text{ReLU}(\cdot) = \max(0, \cdot)$ (Kipf and Welling 2017). $H^{(l)}$ is the matrix of activations in the l th layer, where $H^{(0)} = N$. In addition, since Eq. (3) is essentially equivalent to aggregating node representations from their first-order neighborhood, a GCN is often regarded as a bridge between the spectral-based and spatial-based methods (Zhang et al. 2019).

2.3 Graph Attention Network (GAT)

An attention mechanism can generally be defined as the concentration or processing of information with varying degrees of importance selectively for humans or machines, and it has been successfully employed in numerous deep learning models. Google's

DeepMind team first combined the idea of attentional mechanisms with RNNs for processing image data by adaptively selecting and processing part of the image instead of the image as a whole (Mnih et al. 2014). Bahdanau et al. (2016) proposed the attention coefficients calculation method in the field of neural machine translation, so the model can selectively focus on the important part of the information. Vaswani et al. (2017) further introduced a self-attention mechanism and demonstrated that the self-attention layer is sufficient to construct an efficient model. Inspired by the idea of attention mechanisms, Veličković et al. (2018) introduced an attention-based architecture for graph-structured data node classification, namely, GAT. The shortcomings of previous approaches based on the graph convolutions or its approximations are solved by using a masked self-attention layer. The GAT input is a set of node features (N), and its output is a new set of node features F' . The graph attention layer can be obtained using the following steps:

- (1) Calculating attention coefficients e_{ij} using the below function

$$e_{ij} = \text{attention}(W\vec{n}_i, W\vec{n}_j) = \text{Leaky ReLU}\left(\vec{a}^T [W\vec{n}_i \ W\vec{n}_j]\right), \quad (4)$$

where W is a weight matrix $W \in R^{F' \times F}$; e_{ij} represents the importance of node j 's features to node i ; the attention mechanism a is a single-layer feedforward neural network, parameterized by a weight vector $\vec{a} \in R^{2F'}$; \cdot^T and $\|$ denote transposition and concatenation operations, respectively. The Leaky ReLU(\cdot) is a nonlinear activation function with a negative input slope $\alpha = 0.2$.

- (2) A softmax (\cdot) function is used to normalize the attention coefficients, which makes it easy to compare coefficients between different nodes (Fig. 2a).

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{kj})}. \quad (5)$$

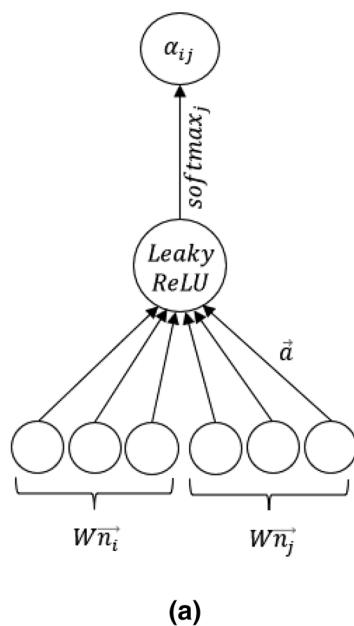
- (3) The final output features are calculated by the linear combination of the normalized attention coefficient and the original node feature vector

$$\vec{n}_i = \sigma \left(\sum_j \alpha_{ij} W\vec{n}_j \right), \quad (6)$$

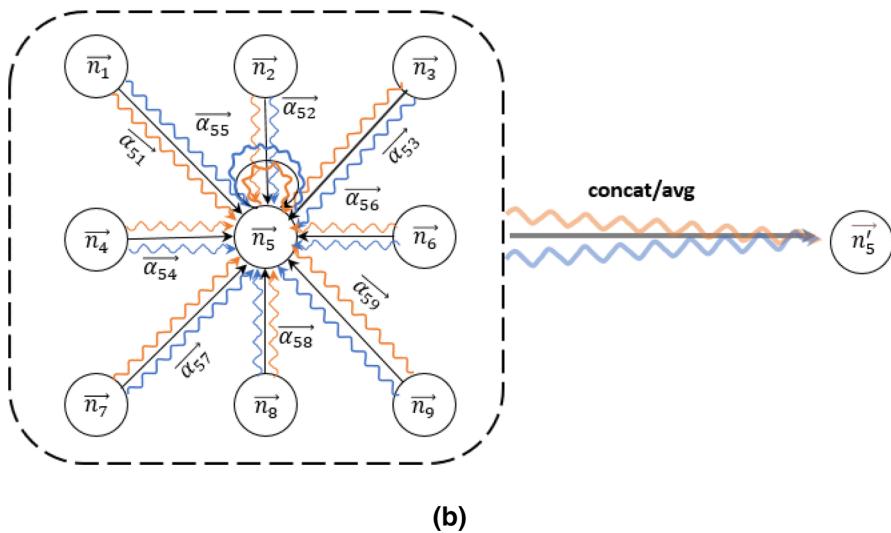
The P independent attention mechanisms for a multi-head attention mechanism implement the transformation of the above equation, and the output can be calculated as follows.

$$\vec{n}_i = \|_{p=1}^P \sigma \left(\sum_j \alpha_{ij}^p W^p \vec{n}_j \right), \quad (7)$$

where α_{ij}^p are normalized attention coefficients computed by the p th attention mechanism (α^p), and W^p is the corresponding input linear transformation's weight matrix.



(a)



(b)

Fig. 2 Graph attention layer: **a** graph attention mechanism $attention(W\vec{n}_i, W\vec{n}_j)$, parameterized by a weight vector $\vec{a} \in R^{2F'}$ using a LeakyReLU function; and **b** multi-head attention (with $P = 6$ heads). Different arrow styles and colors represent independent attention computations

A multi-head attention on the final (prediction) layer of the network can be carried out when averaging the final nonlinearity (Fig. 2b).

$$\vec{n}_i = \sigma \left(\frac{1}{P} \sum_{p=1}^P \sum_j \alpha_{ij}^p W^p \vec{n}_j \right). \quad (8)$$

2.4 Convolutional Neural Networks (CNN)

With improvements in graphics processing unit computing power and various regularization techniques, such as dropout (Srivastava et al. 2014) and batch normalization (Ioffe and Szegedy 2015), CNNs have been widely applied in many fields (e.g., Karimpouli et al. 2020; Liu et al. 2021; Wang et al. 2022). A typical CNN extracts deep information by implementing convolution and pooling operations, and finally maps information to a classification result by fully connected layers (Fig. 3). Compared with the traditional manual features, high-level features extracted by CNNs has been proven superior (Lu et al. 2017). Each node in the output layer represents the credibility of each category in the classification result (Fig. 3). The layers are connected by nonlinear activation functions. Loss functions are objective functions of gradient descent learning in the back propagation. By minimizing the loss function, a reliable predictive model can be established (Krizhevsky et al. 2012; LeCun et al. 2015).

3 Case Study

The data used in this study are collected from Yin et al. (2022) and Zuo et al. (2022), which were compiled from Carranza and Hale (2001, 2002, 2003). These data contain four prospecting information layers, namely, NE- and NW-trending faults, Agno Batholithic pluton margins, and porphyry intrusive contacts (Fig. 4), which have strong spatial correlations with known gold mineralization. Based on these datasets, various statistical models were adopted to generate a mineral potential map (e.g., Carranza and Hale 2001, 2002, 2003; Yin et al. 2022; Zuo et al. 2022). These studies demonstrated that these prospecting information layers can support the production of a prospectivity map associated with gold mineralization. The size of each pixel in this study is 100 m × 100 m. The process of prospecting information layers can be found in Carranza and Hale (2003) and Yin et al. (2022).

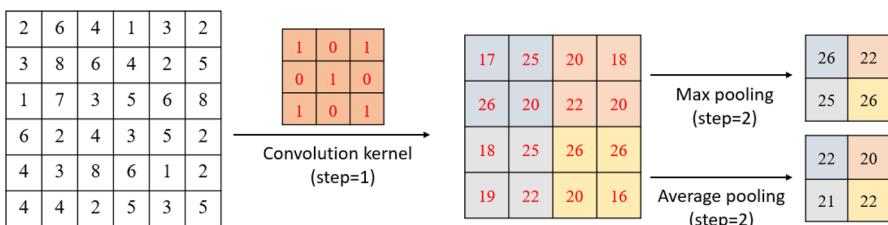


Fig. 3 A schematic diagram of extracting feature maps by a convolutional neural network

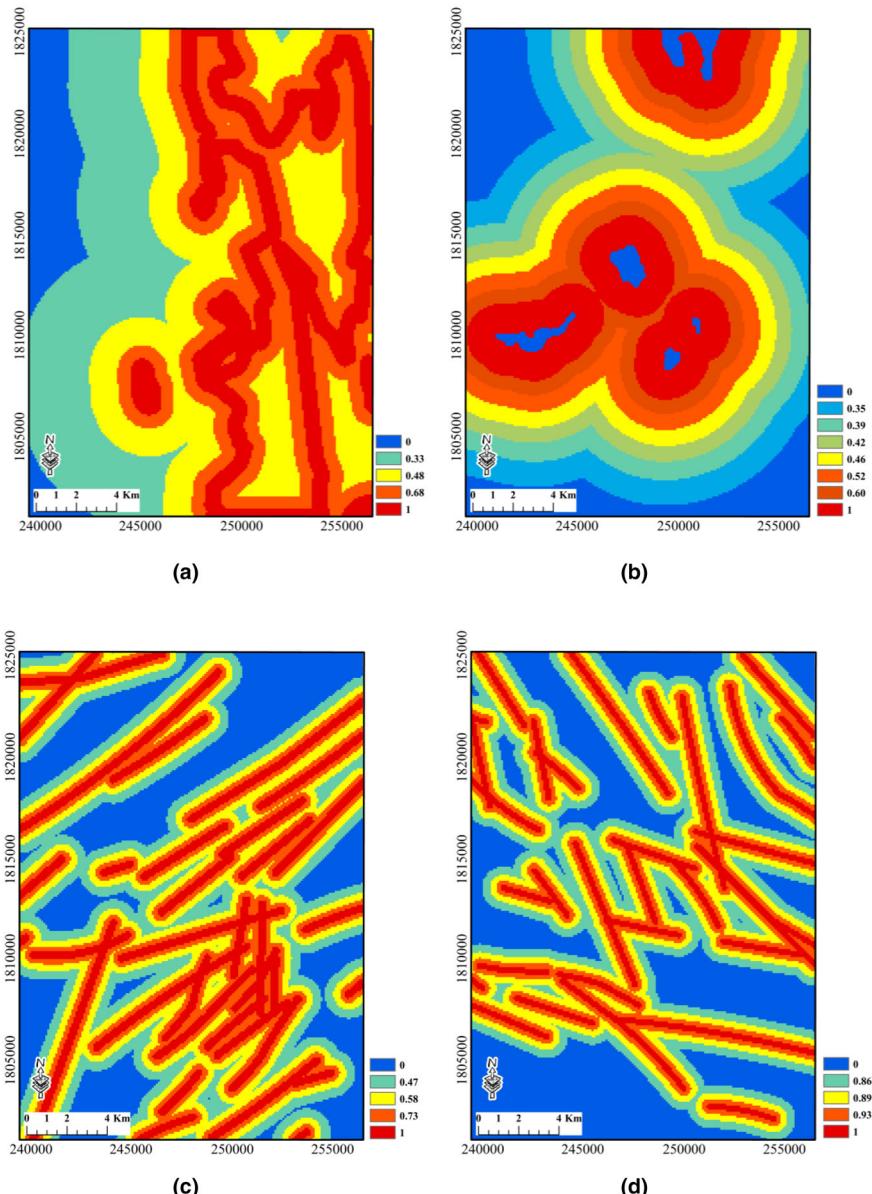
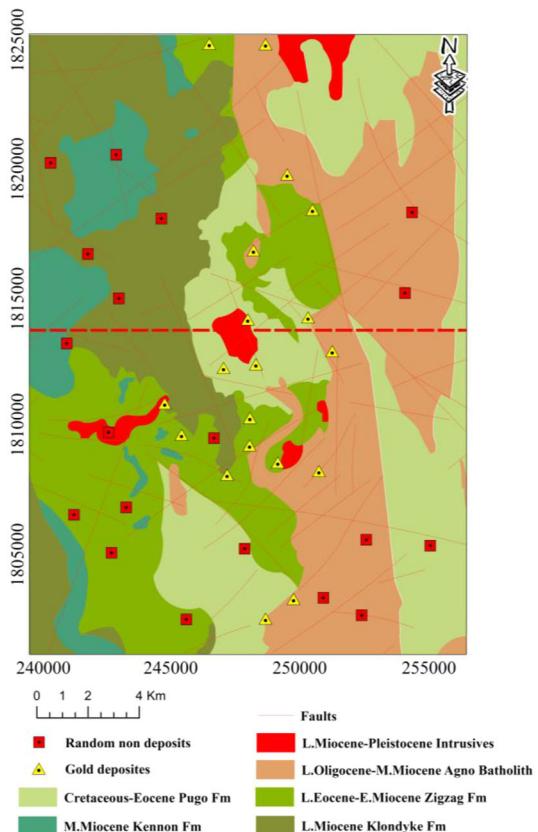


Fig. 4 The input data: **a** Agno Batholithic pluton margins, **b** porphyry intrusive contacts, **c** NE- and **d** NW-trending faults

Fig. 5 A map showing the positive and negative samples and training and testing area. The areas above and below black lines indicate the training and testing areas, respectively. The base map is a potential map produced by Zuo et al. (2022)



The study area was divided into two parts. The lower part is used for training deep learning algorithms, and the remaining part is used for testing the predictive model. The whole study area contains 19 gold deposits. The upper and lower parts contain 7 and 12 gold deposits, respectively. Positive samples are generated by the locations of known mineral deposits, and negative samples are created by random selection in the areas linked to low potential probability indicated by previous studies (e.g., Carranza and Hale 2001, 2002, 2003; Yin et al. 2022; Zuo et al. 2022). The number of positive samples should be the same as that of negative samples to avoid the unbalanced problem. As a result, 12 known gold deposits and 12 random points were selected as positive and negative points in the training area for creating training samples (Fig. 5).

4 Results and Discussion

4.1 GNN Architecture

A GNN was implemented using Python under the PyTorch geometric (PyG) library (<https://pytorch-geometric.readthedocs.io/en/latest/index.html>). To quantitatively compare with the CNN performance, the distance threshold for constructing

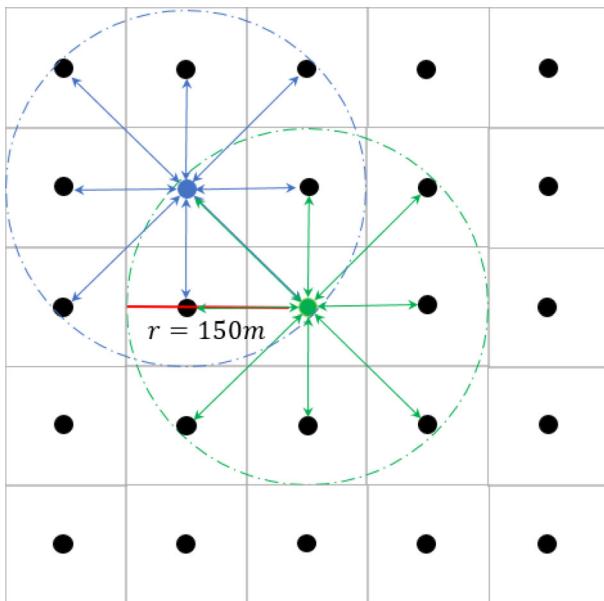


Fig. 6 Topology graph with a 150 m distance threshold. The green and blue nodes are the central node of a topology subgraph; different color arrows represent the edges of different subgraphs; and bidirectional arrows represent undirected graphs

the topology graph was set to 150 m; in other words, each node in a graph can gather information from eight neighboring nodes in the updating process (Fig. 6). All points within a window size of 11×11 are labeled as positive or negative samples to expand the training samples. A total of 1452 (121×12) positive and negative samples were generated, respectively. A total of 80% and 20% positive and negative samples were randomly selected for training and validating the model. To avoid the effects of randomly selected training samples on the predictive results, 10 sets of training and validation samples were selected randomly. The final potential map is the average value of the 10 predictive maps obtained. Both the GCN and GAT consist of four graph convolutional layers and attentional layers, respectively (Fig. 7a). Each layer is activated using a *ReLU* function to avoid the gradient disappearance in the training process. A dropout layer ($P = 0.5$) is added after each layer to prevent model overfitting. Finally, *log_softmax* is used for classification. In a GNN, a negative log likelihood is applied to calculate the loss values (*nll-loss*). The Adam SGD optimizer (Kingma and Ba 2014) is applied to minimize the loss value and optimize the weight vector \vec{w} in GAT and the weight matrix W in GCN and GAT. The W is assessed using their accuracy score in the testing set. In the GNN training process, the iterations and the learning rate were set as 1,000 and 0.001, respectively.

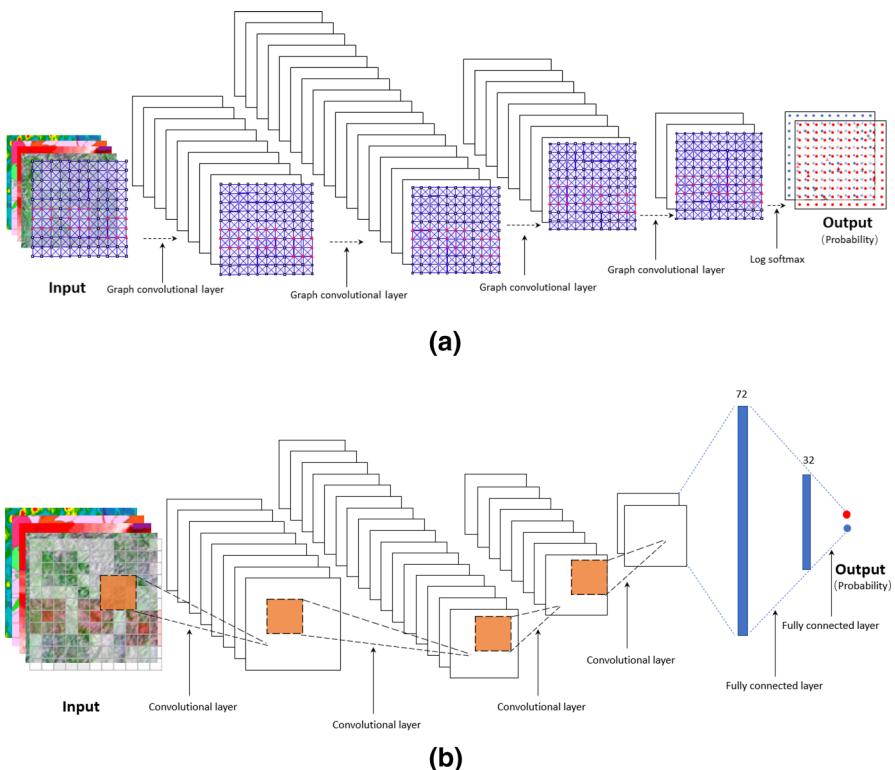


Fig. 7 Architectures of (a) a graph convolutional network and graph attention network, and b convolutional neural network

4.2 CNN Architecture

A CNN was implemented using Python under the torch.nn library (<https://pytorch.org/docs/stable/torch.html>). Considering the size of the input samples, the convolution kernel was set to 3×3 . The convolutional layer was only used to build a CNN (Fig. 7b) to compare the performance of a GNN and a CNN. Each convolutional layer was also activated by the ReLU function and followed by a dropout layer $P = 0.5$. Finally, a linear fully connected layer was used for classification. The cross-entropy was used to calculate the loss values (CrossEntropyLoss), which is the sum of *log_softmax* and *loss value*, and the AdaGrad (Duchi et al. 2011) was applied as an optimizer to update the weights and bias of the CNN.

The training sample size (or patch size) was determined by the nearest distance between a pair of known mineral deposits. Such a method can ensure that each training sample contains only one known mineral deposit. Figure 8 displays the histogram of the nearest distances between a pair of known mineral deposits and summarized statistics. The maximum size of the training samples was set as 11×11 pixels because the nearest distance between a pair of known mineral deposits was 1,100 m. The minimum size of the training samples was set as 6×6 pixels to ensure that each training sample contains

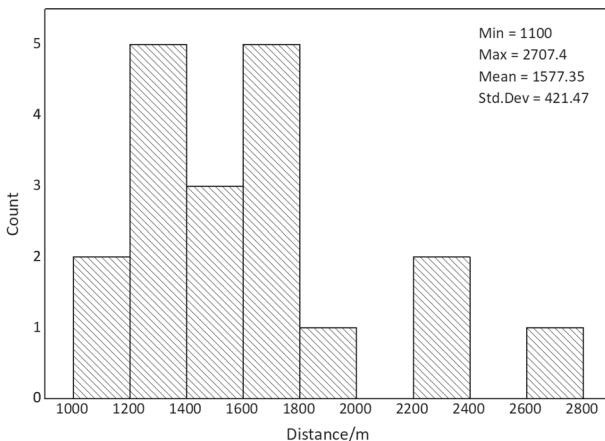


Fig. 8 A histogram of the nearest distances between a pair of known mineral deposits

known positive and negative points. A data augmentation was implemented to create sufficient amount of positive and negative samples (Yang et al. 2022). Patches of 11×11 around the positive and negative samples were cropped, and 12 known positive and negative points were located in the center of these patches. For each patch, 36 different subpatches can be generated with a size of 6×6 pixels without repetition, with the aim of creating the maximum number of training samples. As a result, a total of 432 (36×12) positive and negative samples were created, respectively. A total of 80% positive and negative samples were used for training, and the rest of samples were used for validation. The above process was repeated 10 times to reduce the effects of randomly selected training samples on the predictive results. The final predictive map was the average value of the 10 predictive maps obtained. In the training process of a CNN, the iterations and the learning rate were set as 1,000 and 0.001, respectively.

4.3 Mapping Gold Mineralization Potentials

Figure 9 shows the accuracy and loss curves of the GCN, GAT, GAT* (GAT with the first two layers consisting of $P = 6$ attention heads), and CNN in the training and validation sets. It can be observed that the classification accuracy of a GCN, GAT, GAT*, and CNN reaches 0.972, 0.969, 0.958, and 0.996 on the training datasets, and 0.971, 0.836, 0.948, and 0.994 on the validation datasets after 1,000 iterations, indicating that these four classification models are well trained. Meanwhile, GAT has a high classification accuracy and a low loss value in the training dataset, but it has relatively low classification accuracy and a high loss value in the validation set, suggesting an overfitted GAT model to some extent. Compared with GAT, GAT* has a larger classification accuracy and a smaller loss value in the validation dataset.

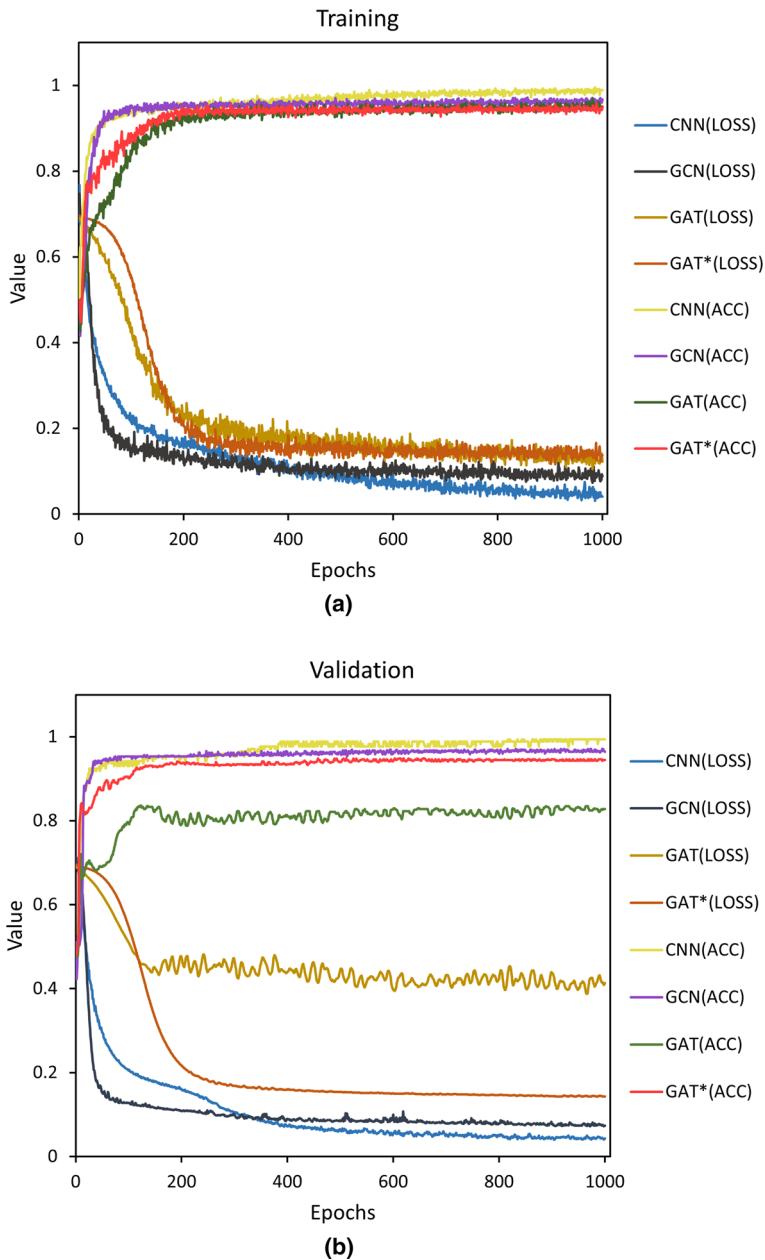


Fig. 9 Accuracy and loss curves of GCN, GAT, GAT*, and CNN in the **a** training and **b** validation sets

The concentration-area fractal model (C-A) (Cheng et al. 1994) was used to classify four mineral potential maps obtained using GCN, GAT, GAT*, and CNN based on ArcFractal (Zuo and Wang 2020) (Figs. 10 and 11). The C-A model can determine robust cut-off values to separate a potential map into several regions in terms of the log–log relations between the probability versus the cumulative area. Most of the known mineral deposits were located in or near the areas linked to high probability. Additionally, the obtained potential maps had spatial patterns similar to those of previous studies using pixel-wise approaches, which include other deep learning algorithms (Yin et al. 2022; Zuo et al. 2022) and traditional methods (Carranza and Hale 2001, 2002, 2003), suggesting good performance of GNNs and CNN in MPM. Table 1 lists a statistics of the cumulative areas versus the cumulative number of mineral deposits. A total of 5% of the predictive areas delineated by the GCN, GAT, GAT*, and CNN contain 26.32%, 21.05%, 26.32%, and 10.53% of the total number of mineral deposits, indicating that GNNs, including the GCN, GAT and GAT*, have better performance than a CNN in the identification of high-probability regions locating mineral deposits.

A true/false prediction rate plot (Talebi et al. 2022) was used to evaluate the performance of MPM models. The relationships between the proportion of mineral deposits and non-deposits versus the percentage of areas classified as prospective areas were plotted simultaneously to show the curves of true and false positive rates, respectively. This plot can evaluate the ability of the obtained mineral potential maps from the aspect of discovering new mineral deposits and avoiding false-positive predictions to reduce exploration cost by limiting the extent of the areas with potentials as much as

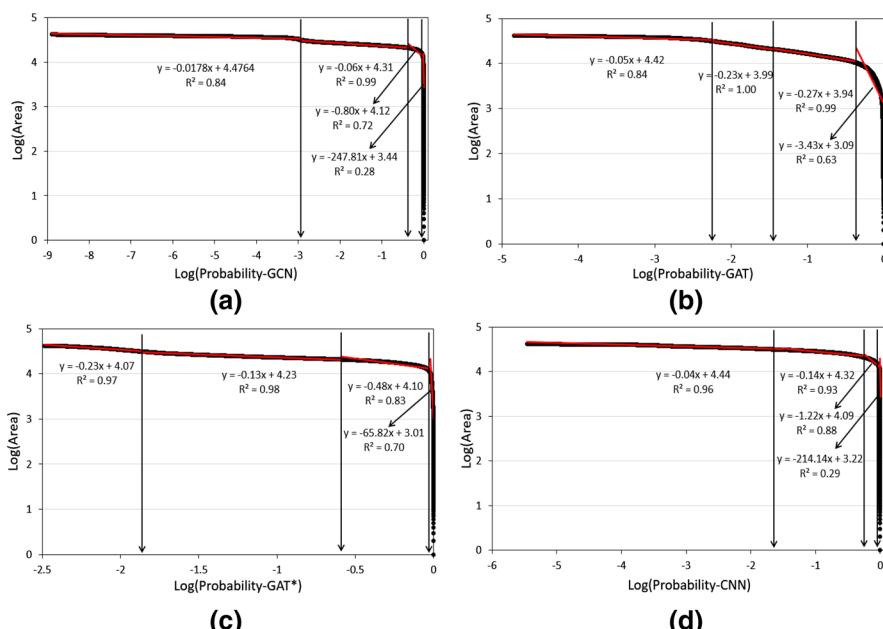


Fig. 10 Log–log plots of area versus the predictive probability obtained by **a** GCN, **b** GAT, **c** GAT*, and **d** CNN

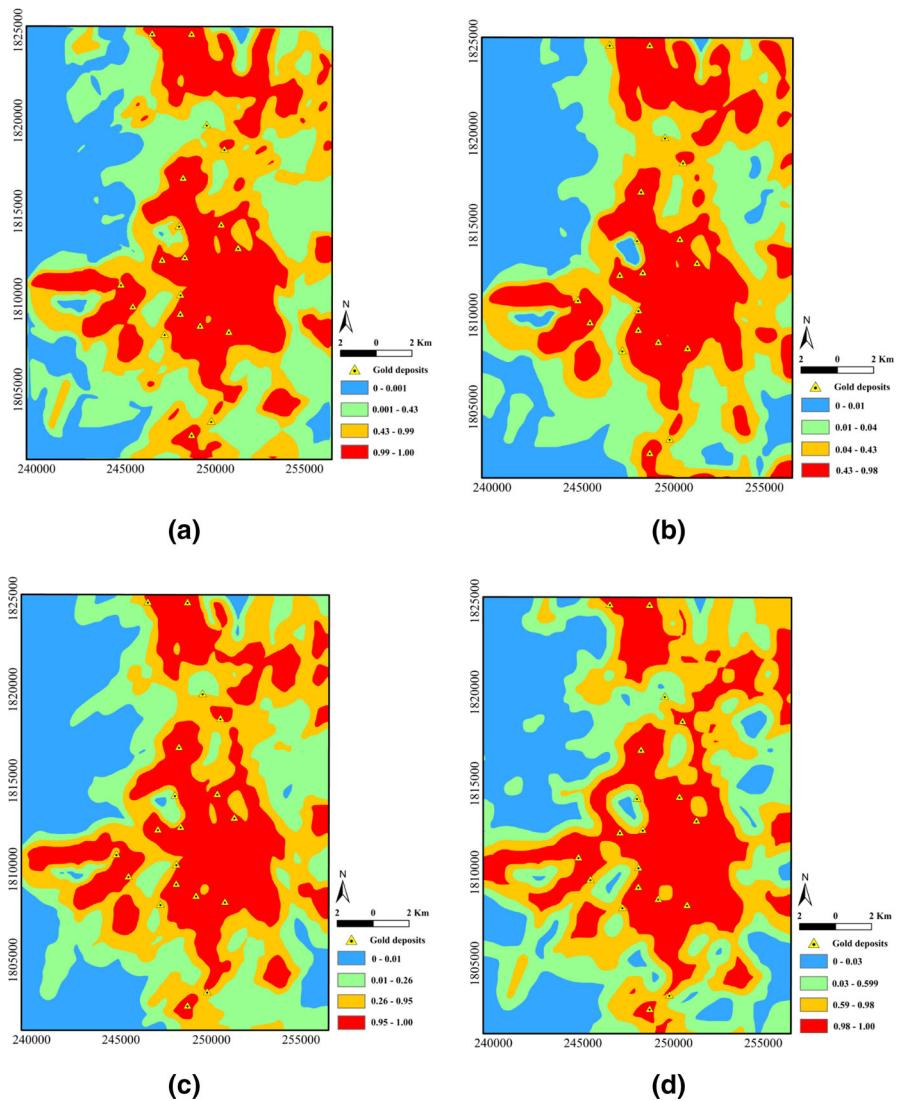
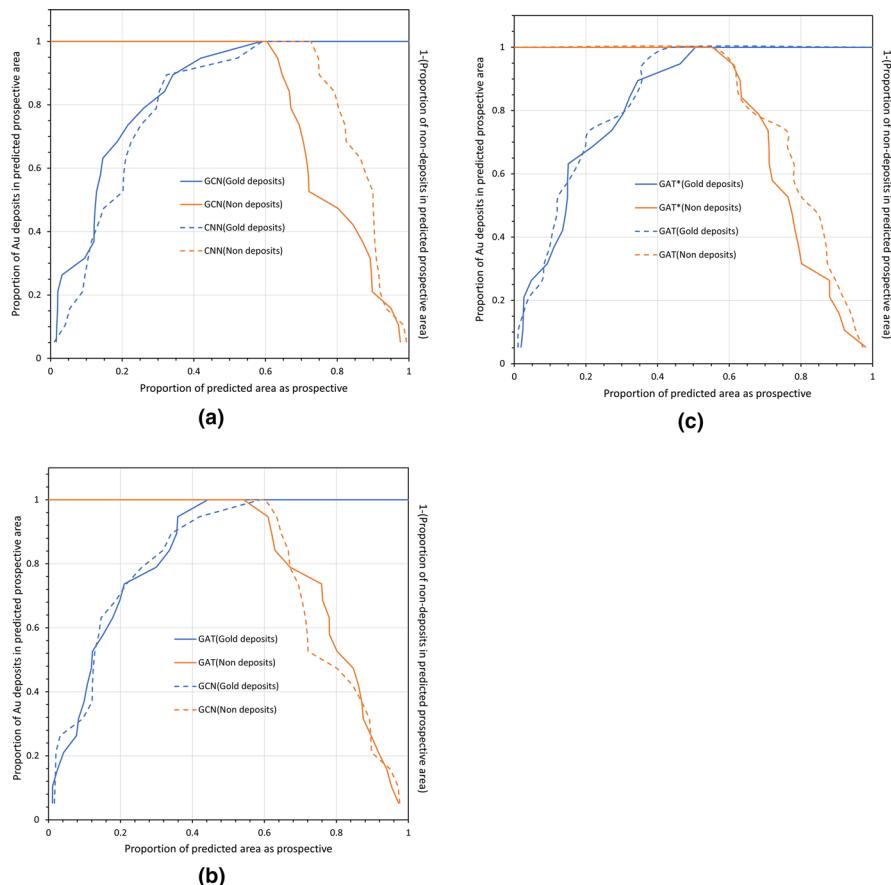


Fig. 11 Mineral potential maps obtained by **a** GCN, **b** GAT, **c** GAT*, and **d** CNN

possible. Figure 12 displays the true/false prediction rate plots to compare the performance of GCN versus CNN, GAT versus GCN, and GAT versus GAT*. It can be observed that the GCN performs better than the CNN in predicting positive samples (Fig. 12a) since GNNs can capture the spatial pattern of prospecting information from the whole topology graph, while CNNs only capture the geospatial patterns from a fixed size of a pixel patch. However, GCN performs worse than CNN in avoiding false predictions (Fig. 11a). In Fig. 11b, 44.27% and 58.82% of the prospective areas identified by GAT and GCN, respectively, can cover all the known mineral deposits

Table 1 Proportion of mineral deposits within 5%, 10%, and 20% of predictive regions identified by GCN, GAT, GAT*, and CNN

Models	5% of the total study area	10% of the total study area	20% of the total study area
GCN	5/19 (26.32%)	6/19 (31.58%)	13/19 (68.422%)
GAT	4/19 (21.05%)	7/19 (36.84%)	13/19 (68.422%)
GAT*	5/19 (26.32%)	6/19 (31.58%)	12/19 (63.16%)
CNN	2/19 (10.53%)	5/19 (26.32%)	9/19 (47.37%)

**Fig. 12** True/false prediction rate plots of **a** GCN versus CNN, **b** GAT versus GCN, and **c** GAT versus GAT*

(Fig. 12b). This indicates better performance of GAT than GCN because the attention mechanism can improve the representation ability of GNNs by specifying different weights to different adjacent nodes of a centered node when aggregating information. The performance of GAT and GAT* is comparatively the same as shown in Fig. 12c. It is worth noting that the multi-head attention mechanism used in GAT* allows the model to extract information from different representation subspaces concentrating on different parts of the input data in parallel, which can help improve the representation ability compared to a single head in theory. The attention mechanism in MPM should be further studied in the future.

5 Conclusions

Traditional mineral prospectivity mapping was implemented based on a single pixel and may ignore the spatial patterns associated with mineralization. Although a convolutional neural network is an image-based approach that involves the spatial structure of prospecting information in the Euclidean domain, it has a fixed size of the inputted image and does not well consider the spatial anisotropy of mineralization. In this study, graph deep learning algorithms were introduced to capture the spatial patterns of prospecting information in mineral prospectivity mapping. A comparative study of graph deep learning algorithms with a convolutional neural network is conducted. The cumulative areas versus the cumulative number of mineral deposits of different methods suggest that GNNs have better performance than a CNN in the identification of high-probability regions locating mineral deposits. In addition, the true/false prediction rate plots for the four potential maps obtained using GNNs and CNN indicate that GCN performs better than the CNN in predicting positive samples. The following conclusions were obtained: (1) geological prospecting data can be regarded as graphs that have a strong ability to capture the spatial intercorrelations between locations of mineralization and prospecting information, (2) graph deep learning algorithms are powerful tools for producing mineral potential maps and more studies related to the application of graph deep learning algorithms in mineral prospectivity mapping are expected in the future, and (3) the application of attention mechanisms is a frontier field and should be further explored in mineral prospectivity mapping.

Acknowledgements We are grateful for two reviewers' comments and suggestions, which improved this paper. This study was supported by the National Natural Science Foundation of China (Nos. 41972303 and 42172326).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Agterberg FP (1989) Computer programs for mineral exploration. *Science* 245:76–81. <https://doi.org/10.1126/science.245.4913.76>

- Ashish V, Noam S, Niki P, Jakob U, Llion J, Aidan NG, Lukasz K, Illia P (2017) Attention is all you need. *Adv Neural Inform Process Syst* 30:445
- Bahdanau D, Cho K, Bengio Y (2016) Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473)
- Bonham-Carter GF (1994) Geographic information systems for geoscientists: modelling with GIS. Pergamon Press, p 398
- Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: going beyond euclidean data. *IEEE Signal Process Mag* 34:18–42. <https://doi.org/10.1109/MSP.2017.2693418>
- Carranza EJM (2008) Geochemical anomaly and mineral prospectivity mapping in GIS: Amsterdam. In: Hale M (ed) Handbook of exploration and environmental geochemistry. Elsevier, New York, p 351
- Carranza EJM, Laborte AG (2015) Data-driven predictive mapping of gold prospectivity, Baguio district, Philippines: application of random forests algorithm. *Ore Geol Rev* 71:777–787. <https://doi.org/10.1016/j.oregeorev.2014.08.010>
- Carranza EJM, Hale M (2001) Logistic regression for geologically constrained mapping of gold potential, Baguio district, Philippines. *Explor Min Geol* 10:165–175. <https://doi.org/10.2113/0100165>
- Carranza EJM, Hale M (2003) Evidential belief functions for data-driven geologically constrained mapping of gold potential, Baguio district, Philippines. *Ore Geol Rev* 22:117–132. [https://doi.org/10.1016/S0169-1368\(02\)00111-7](https://doi.org/10.1016/S0169-1368(02)00111-7)
- Carranza EJM, Hale M (2002) Where are porphyry copper deposits spatially localized? A case study in Benguet province, Philippines. *Nat Resour Res* 11:45–59. <https://doi.org/10.1023/A:1014287720379>
- Carranza EJM, Mangaang JC, Hale M (1999) Application of mineral exploration models and GIS to generate mineral potential maps as input for optimum land-use planning in the Philippines. *Nat Resour Res* 8:165–173. <https://doi.org/10.1023/A:1021846820568>
- Chen J, Jiao L, Liu X, Li L, Liu F, Yang S (2022) Automatic graph learning convolutional networks for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 60:1–16. <https://doi.org/10.1109/TGRS.2021.3135084>
- Cheng Q, Agterberg FP, Ballantyne SB (1994) The separation of geochemical anomalies from background by fractal methods. *J Geochem Explor* 51:109–130. [https://doi.org/10.1016/0375-6742\(94\)90013-2](https://doi.org/10.1016/0375-6742(94)90013-2)
- Du X, Zheng X, Lu X, Doudkin AA (2021) Multisource remote sensing data classification with graph fusion network. *IEEE Trans Geosci Remote Sens* 59:10062–10072. <https://doi.org/10.1109/TGRS.2020.3047130>
- Duchi JC, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res* 12:2121–2159. <https://doi.org/10.5555/1953048.2021068>
- Fu S, Liu W, Zhang K, Zhou Y (2021) Example-feature graph convolutional networks for semi-supervised classification. *Neurocomputing* 461:63–76. <https://doi.org/10.1016/j.neucom.2021.07048>
- Gallicchio C, Micheli A (2010) Graph echo state networks. *IEEE Int Joint Conf Neural Netw*. <https://doi.org/10.1109/IJCNN.2010.5596796>
- Gori M, Monfardini G, Scarselli F (2005) A new model for learning in graph domains. *IEEE Int Joint Conf Neural Netw*. <https://doi.org/10.1109/IJCNN.2005.1555942>
- Guan Q, Ren S, Chen L, Yao Y, Hu Y, Wang R, Feng B, Gu L, Chen W (2022) Recognizing multivariate geochemical anomalies related to mineralization by using deep unsupervised graph learning. *Nat Resour Res*. <https://doi.org/10.1007/s11053-022-10088-x>
- Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Paper presented at the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015. <https://doi.org/10.48550/arXiv.1502.03167>
- Karimpouli S, Tahmasebi P, Saenger EH (2020) Coal cleat/fracture segmentation using convolutional neural networks. *Nat Resour Res* 29:1675–1685. <https://doi.org/10.1007/s11053-019-09536-y>
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Adv neural inf proc syst pp 1097–1105. <https://doi.org/10.1145/3065386>
- Kingma D, Ba J (2014) Adam: a method for stochastic optimization. <https://doi.org/10.48550/arXiv.1412.6980>
- Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. arXiv preprint. <https://doi.org/10.48550/arXiv.1609.02907>
- Li C, Qin X, Xu X, Yang D, Wei G (2020a) Scalable graph convolutional networks with fast localized spectral filter for directed graphs. *IEEE Access* 8:105634–105644. <https://doi.org/10.1109/ACCESS.2020.3009520>

- Li S, Chen J, Xiang J (2020b) Applications of deep convolutional neural networks in prospecting prediction based on two-dimensional geological big data. *Neural Comput Appl* 32:2037–2053. <https://doi.org/10.1007/s00521-019-04341-3>
- Li S, Chen J, Liu C, Wang Y (2021a) Mineral prospectivity prediction via convolutional neural networks based on geological big data. *Journal of Earth Science* 32:327–347. <https://doi.org/10.1007/s12583-020-1365-z>
- Li T, Zuo R, Xiong Y, Peng Y (2021b) Random-drop data augmentation of deep convolutional neural network for mineral prospectivity mapping. *Nat Resour Res* 30:27–38. <https://doi.org/10.1007/s11053-020-09742-z>
- Li T, Zuo R, Zhao X, Zhao K (2022) Mapping prospectivity for regolith-hosted REE deposits via convolutional neural network with generative adversarial network augmented data. *Ore Geol Rev* 142:104693. <https://doi.org/10.1016/j.oregeorev.2022.104693>
- Liu Y, Zhang ZL, Liu X, Xia WL, XH, (2021) Deep learning-based image classification for online multi-coal and multi-class sorting. *Comput Geosci* 157:104922. <https://doi.org/10.1016/j.cageo.2021.104922>
- Lu X, Zheng X, Yuan Y (2017) Remote sensing scene classification by unsupervised representation learning. *IEEE Trans Geosci Remote Sens* 55:5148–5157. <https://doi.org/10.1109/TGRS.2017.2702596>
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436. <https://doi.org/10.1038/nature14539>
- Mnih V, Heess N, Graves A, Kavukcuoglu K (2014) Recurrent models of visual attention. *Adv Neural Inform Process Syst* 27:7789
- Porwal A, Carranza EMJ (2015) Introduction to the special issue: GIS-based mineral potential modelling and geological data analyses for mineral exploration. *Ore Geol Rev* 71:477–483. <https://doi.org/10.1016/j.oregeorev.2015.04.017>
- Rodriguez-Galiano VF, Chica-Olmo M, Chica-Rivas M (2014) Predictive modelling of gold potential with the integration of multisource information based on random forest: a case study on the Rodalquilar area, Southern Spain. *Int J Geogr Inf Sci* 28:1336–1354. <https://doi.org/10.1080/13658816.2014.885527>
- Scarselli F, Gori M, Tsai AC, Hagenbuchner M, Monfardini G (2009) Computational capabilities of graph neural networks. *IEEE Trans Neural Netw* 20:81–102. <https://doi.org/10.1109/TNN.2008.2005141>
- Sperduti A, Starita A (1997) Supervised neural networks for the classification of structures. *IEEE Trans Neural Netw* 8:714–735. <https://doi.org/10.1109/72.572108>
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958. <https://doi.org/10.5555/2627435.2670313>
- Sun T, Li H, Wu K, Chen F, Zhu Z, Hu Z (2020) Data-driven predictive modelling of mineral prospectivity using machine learning and deep learning methods: a case study from Southern Jiangxi Province, China. *Minerals* 10:102. <https://doi.org/10.3390/min10020102>
- Talebi H, Mueller U, Peeters LJM, Otto A, de Caritat P, Tolosana-Delgado P, van den Boogaart KG (2022) Stochastic modelling of mineral exploration targets. *Math Geosci* 54:593–621. <https://doi.org/10.1007/s11004-021-09989-z>
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Opolsukhin I (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. ed. by Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R. (Curran Associates, Red Hook, 2017), pp 5998–6008
- Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. arXiv preprint. <https://doi.org/10.48550/arXiv.1710.10903>
- Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ (1989) Phoneme recognition using time-delay neural networks. *IEEE Trans Acoust Speech Signal Process*. <https://doi.org/10.1109/29.21701>
- Wang X, Zuo R, Wang Z (2022) Lithological mapping using a convolutional neural network based on stream sediment geochemical survey data. *Nat Resour Res*. <https://doi.org/10.1007/s11053-022-10096-x>
- Wang Z, Zuo R (2022) Mineral prospectivity mapping using a joint singularity-based weighting method and long short-term memory network. *Comput Geosci* 158:104974. <https://doi.org/10.1016/j.cageo.2021.104974>
- Wu L, Cui P, Pei J, Zhao L, Song L (2022) Graph Neural Networks. In: Wu L, Cui P, Pei J, Zhao L (eds) Graph neural networks: foundations, frontiers, and applications. Springer, Singapore. <https://doi.org/10.1007/978-981-16-6054-23>
- Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32:4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>

- Xiong Y, Zuo R, Carranza EJM (2018) Mapping mineral prospectivity through big data analytics and a deep learning algorithm. *Ore Geol Rev* 102:811–817. <https://doi.org/10.1016/j.oregeorev.2018.10.006>
- Yang N, Zhang Z, Yang J, Hong Z (2022) Applications of data augmentation in mineral prospectivity prediction based on convolutional neural networks. *Comput Geosci* 161:105075. <https://doi.org/10.1016/j.cageo.2022.105075>
- Yang N, Zhang Z, Yang J, Hong Z, Shi S (2021) A convolutional neural network of GoogLeNet applied in mineral prospectivity prediction based on multi-source geoinformation. *Nat Resour Res* 30:3905–3923. <https://doi.org/10.1007/s11053-021-09934-1>
- Yin B, Zuo R, Xiong Y (2022) Mineral prospectivity mapping via gated recurrent unit model. *Nat Resour Res* 31:2065–2079. <https://doi.org/10.1007/s11053-021-09979-2>
- Zhang S, Carranza EJM, Wei H, Xiao K, Yang F, Xiang J, Zhang S, Xu Y (2021) Data-driven mineral prospectivity mapping by joint application of unsupervised convolutional auto-encoder network and supervised convolutional neural network. *Nat Resour Res* 30:1011–1031. <https://doi.org/10.1007/s11053-020-09789-y>
- Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. *Comput Soc Netw* 6:11. <https://doi.org/10.1186/s40649-019-0069-y>
- Zuo R (2020) Geodata science-based mineral prospectivity mapping: a review. *Nat Resour Res* 29:3415–3424. <https://doi.org/10.1007/s11053-020-09700-9>
- Zuo R, Wang J (2020) ArcFractal: an ArcGIS add-in for processing geoscience data using fractal/multifractal models. *Nat Resour Res* 29:3–12. <https://doi.org/10.1007/s11053-019-09513-5>
- Zuo R, Luo Z, Xiong Y, Yin B (2022) A geologically constrained variational autoencoder for mineral prospectivity mapping. *Nat Resour Res* 31:1121–1133. <https://doi.org/10.1007/s11053-022-10050-x>

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.