Autocompletion of Design Data in Semantic Building Models using Link Prediction and Graph Neural Networks

Viktor Eisenstadt[1], *Jessica Bielski*[2], *Christoph Langenhan*[3], *Klaus-Dieter Althoff*[4], *Andreas Dengel*[5]
[1,4,5]*German Research Center for Artificial Intelligence (DFKI)*
[2,3]*Technical University of Munich, Germany*
[1,4]*University of Hildesheim, Germany*
[1,4,5]*{viktor.eisenstadt|klaus-dieter.althoff|andreas.dengel}@dfki.de*
[2,3]*{jessica.bielski|christoph.langenhan}@tum.de*

This paper presents an approach for AI-based autocompletion of graph-based spatial configurations using deep learning in the form of link prediction through graph neural networks. The main goal of the research presented is to estimate the probability of connections between the rooms of the spatial configuration graph at hand using the available semantic information. In the context of early design stages, deep learning-based prediction of spatial connections helps to make the design process more efficient and sustainable using the past experiences collected in a training dataset. Using the techniques of transfer learning, we adapted methods available in the modern graph-based deep learning frameworks in order to apply them for our autocompletion purposes to suggest possible further design steps. The results of training, testing, and evaluation showed very good results and justified application of these methods.

Keywords: *Spatial Configuration, Autocompletion, Link Prediction, Deep Learning.*

INTRODUCTION

Artificial intelligence (AI) is a ubiquitous modern computational technology represented in many domains of the industry and everyday life with services such as automatic translation or completion of textual sentences on mobile devices. In architecture, however, AI is still looking for an opportunity to become widely used, being currently limited to academic research and practically absent in the established design software. In order to raise awareness for the potential of AI in architectural design, the research project metis-II, funded by the German Research Foundation (DFG), investigates the use of AI methods such as deep learning (DL) or case-based reasoning (CBR) for autocompletion of spatial configurations using graph-based semantic building fingerprints and their tensor derivations which turn the graph into a numerical representation for deep learning purposes (see Figure 1) (Eisenstadt et al. 2021). DL-based autocompletion methods are developed to enrich the early conceptual phases when the ideas for the future design are vague and incomplete. The architect can benefit from an AI-based application that is able to suggest missing parts in the current spatial configuration, improving the quality of the designs and the future built environment.

Figure 1
An example of a spatial configuration graph converted into a representation usable by deep learning methods. Using a numerical typology for the semantic entities (i.e., *room* and *connection*), the graph is converted into a one-hot encoded tensor.

Based on design experiences of already designed and built architecture stored in a dataset, artificial neural networks (ANN) are trained to produce such suggestions, derived from the architectural practice of consulting reference buildings.

In this paper, we present an autocompletion approach in the form of link prediction (LP), an established computational method for graphs to predict the possible direct linkages within a group of nodes. LP is a method widely applied in online recommender systems or social networks to find relations between items or users in order to recommend related products or make friend or networking suggestions. The main goal of the approach of this paper is to predict connections (relations) between rooms (nodes) in a semantic spatial configuration graph (see Figure 1).

Using the deep learning methods of graph neural networks (GNN) it is possible to learn the semantics and architectural contexts of spatial configurations in the provided dataset and predict the probability of connections between the rooms not yet interconnected. The architect benefits from link prediction by instantly receiving connection suggestions, based on the references and experiences recorded by architects in the dataset of the GNN, which can be set to learn various spatial contexts (e.g., flexible or temporary spaces by learning the potential connections of different configurations for the same floor plan).

LP-based autocompletion is included in the autocompletion framework in development for metis-II, where cluster completion and design phase, intention and step prediction complement LP (see Conclusion and Future Work section).

BACKGROUND AND RELATED WORK

The historical background of the work presented in this paper is the research conducted for the *metis* projects whose goal is to support the early stages of architectural design, namely ideation and schematic design, using the modern methods of artificial intelligence. Starting with CBR-based retrieval of helpful past architectural design references and continuing with data augmentation to acquire more architecturally consistent data to learn from, the current iteration of the research projects moved on to the autocompletion of spatial configurations in order to provide methods for efficient human-computer co-creative design process.

Link prediction is a method that provides two evidential aspects to be applied for such autocompletion. Firstly, it is *established in the graph theory* field and so can be seen as a basic method, but also future-proof as its fields of application grow consistently (e.g., in social networks). Secondly, provided that sufficient contextual information on the graph entities is available, link prediction is *compliant with the requirements* of the architectural design domain as the probability of connection can be estimated based on the architectural semantics learned from the labels and attributes of the spatial graph.

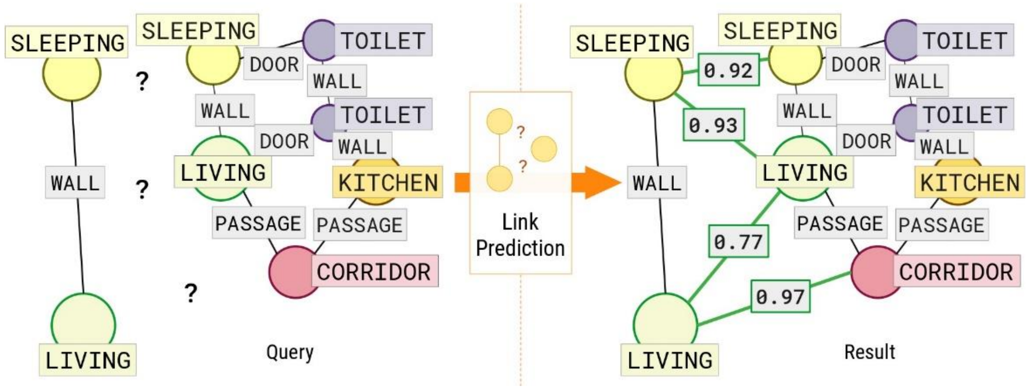


Figure 2
An overview of the link prediction problem for semantic spatial configuration graphs.

Considering only the graph theory aspect, several existing non-AI-based methods could be applied for LP (Liben-Nowell and Kleinberg 2007; Zhou et al. 2009). However, the main problem of all those methods is that they are limited to the single graph information, and neither possess, nor make use of the knowledge of the entirety of the dataset the graphs are contained in. That is, the latent domain-specific information about the architectural design context (e.g., residential housing) cannot be applied to predict relations, potentially resulting in less intelligent predictions. To overcome such problems, deep learning can be applied to learn the architectural contexts by processing the relevant semantic information, as investigated in our previous work (Eisenstadt et al. 2021). The currently emerging field of graph neural networks seemed like an obvious choice to apply DL for link prediction in architecture.

Graph neural networks are a specific type of artificial neural networks that operate directly on graph-based data representations instead of sequence-based data like the recurrent neural networks or multi-dimensional tensors like the convolutional neural networks. Operating directly on the spatial graphs, e.g., for accessibility or adjacency, allows for an unbiased access to the semantics of the graph, while the conversion of the graphs to other data representations (e.g., simple or weighted matrices) might result in loss of relevant information. GNNs have already been applied to investigate deep learning models for link prediction, for example, the approaches Graph Autoencoder (GAE) (Kipf and Welling 2016) and SEAL (Zhang and Chen 2018).

LINK PREDICTION APPROACH

The autocompletion approach presented in this paper applies graph neural networks to predict the existence of semantic relations (i.e., connections such as *Door* or *Passage*) between the nodes (i.e., room types such as *Living* or *Sleeping*) that are not yet connected with each other within the spatial configuration graph. Figure 2 shows an overview of the autocompletion by link prediction: missing links between two separated parts of the spatial configuration graph on the left side should be predicted, the result of the autocompletion process on the right side suggests the percentage of probability of a connection between the rooms.

To investigate the GNNs for our autocompletion purposes, the framework *DGL* (Deep Graph Library) (Wang et al. 2019) was selected. This framework was developed specifically to provide an out-of-the-box implementation of machine learning methods to solve the graph theory problems, including the link prediction problem. In contrast to the other currently available general-purpose frameworks, such as TensorFlow, Keras or PyTorch, DGL has a clear field of application and thus, can be seen as the current standard graph DL library.

In the next sections, the essential components of the DGL-based link prediction approach for graph-based spatial configurations will be presented: the general mode of operation, the model of the neural network, the dataset, and the results of training, testing and evaluation.

Mode of Operation

The general structure of the GNN-based autocompletion approach follows the mode of operation proposed by the DGL developers for the link prediction problem (DGL Team 2018). It is based on *negative sampling*, a technique to maximize similarity value between objects in the similar context in contrast to the objects from different contexts. Most famously it is used in the natural language processing (NLP) library word2vec, more exactly in its Skip-gram model, where it is used to estimate probability of a word to appear in a certain context (i.e., sentence) (Mikolov et al. 2013). Transferred to graph machine learning, this means that the neighboring nodes (i.e., those connected to each other) will have a higher similarity value in relation to their context, in contrast to the nodes not connected to each other in the same context. The corresponding GNN model learns this contextual similarity and classifies a connection between such neighboring nodes as highly probable. In a semantic spatial configuration graph, nodes represent the rooms; accordingly, the graph neural network model should be able to estimate if there is a probability of connection between the rooms based on the learned semantics.

Dataset

In order to provide the GNN with the relevant semantics to learn from and predict the connection correctly, a dataset that consists of such semantic information on available connections of all available graphs is the most essential requirement. For the *metis* research projects, a dataset of semantic spatial configurations for residential housing was previously developed and all graphs were validated for architectural consistency using a specific consistency checker tool (Arora et al. 2021). For GNN-based link prediction, all room connection pairs (e.g., *Living <- Door -> Corridor*) from all graphs were extracted, validated for use in the dataset, and enriched with specific semantic features to provide the GNN with *relevant information to learn from* and assess the contextual similarity correctly.

Following semantic features were added to the connections and their corresponding rooms in order to add sufficient contextual information. Each room got *three additional features* assigned as semantic addition to its current room type label (e.g., *Living* or *Sleeping*):

1. The *housing category class* that describes the number of habitable spaces (i.e., *Sleeping*, *Living*, *Children*, and *Working*) and if the spatial configuration is *open* or *closed*, identified by whether there is a passage between the kitchen and the living room. An example of such a category class is *"2_open"*.
2. The *number of accessibility connections* (*Door*, *Passage*, *Entrance*, or *Stairs*) to reach the room from the corridor which is usually a central circulation and access point to other spaces. This feature was inspired by the *"Step depth"* measurement described in *SpaceSyntax* (UCL Space Syntax 2022).
3. *Connectivity grade*: the number of other rooms accessible from this room via accessibility connections, calculated using the established Dijkstra shortest path algorithm (Dijkstra 1959).

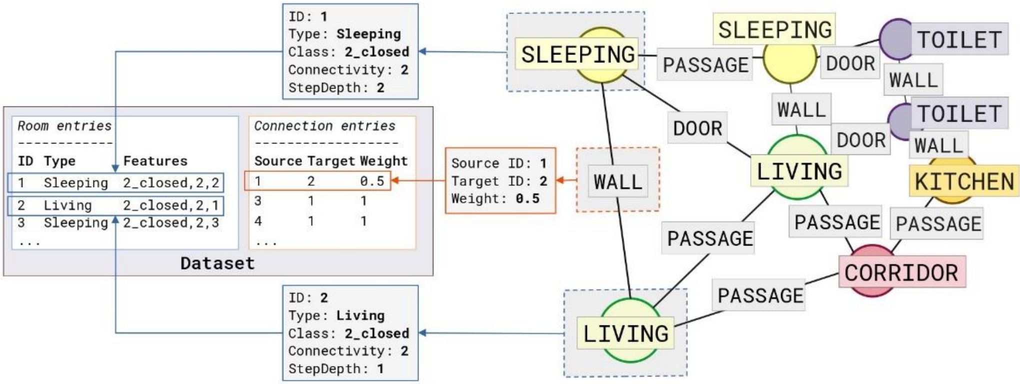


Figure 3
An example of a connection entry and its corresponding room entries in the dataset.

Each connection edge was further enriched with the *weight* specific for the connection type. For example, 1 for the accessibility connections (see above) and 0.5 for the adjacency connections (*Wall*, *Slab*, *Window*).

Following requirements were then applied to validate the inclusion of the connection into the link prediction dataset:

- The spatial configuration that contains the connection should include *at least one Corridor* to ensure the application of Space Syntax' "Step depth".
- The spatial configuration that contains the connection *should not have the inaccessible spaces*, i.e., those that have no accessibility via an open connection of type *Door*, *Passage*, *Entrance*, or *Stairs*.

To this end, the dataset eventually consisted of two separate entry lists (see Figure 3):

- *Connection entries* with *IDs* of the connected rooms and the *weight* of the connection type
- *Room entries* linked with connection entries via IDs and furthermore consisting of the *type of the room* (label) and the corresponding additional *semantic features* (see above)

While features and type labels were initialized as *categorical data* using the binning methods (Lutes 2021), all other data was used in the *numerical* form. Entry examples are shown in Figure 3.

Model

The actual graph neural network model used for link prediction follows loosely the model structure proposed by the Deep Graph Library developers in which the layers of the GNN learn positive and negative (i.e., existing and non-existing) connection examples. DGL's link prediction methods apply the approach *SEAL* (Zhang and Chen 2018), which is based on the extraction and heuristic evaluation of the surrounding semantic information of the investigated connection. More specifically, SEAL evaluates the features of the nodes and edges of the connection's local enclosing subgraphs of different connection depths (hops), counting the depth from the source, as well as from the target of the connection. The SEAL-based GNN is then able to learn the contextual graph structure using the features for *explicit* (semantics and labels) and *latent* (here: residential housing patterns) information encoded in the connections. Figure 4 shows the SEAL-based graph neural network model for link prediction in spatial configurations.

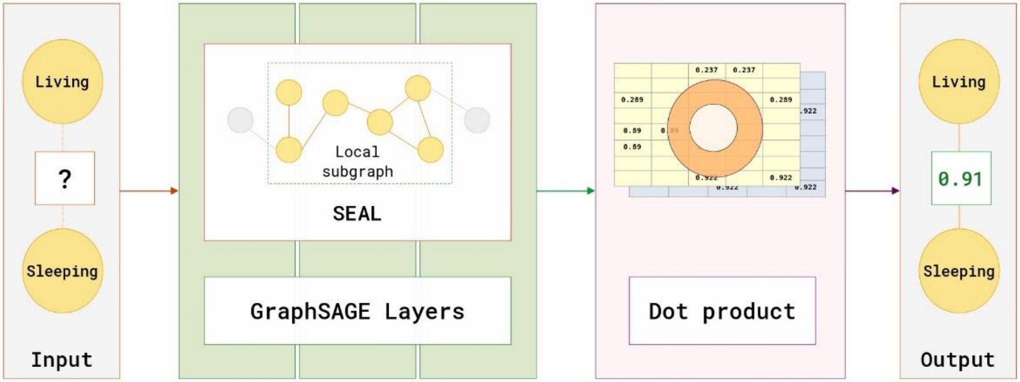


Figure 4
The GNN Model used for link prediction learns local subgraphs (middle blocks) and produces a prediction score (right block) for the query connection (left block).

Historically, SEAL is the extension of WLNM (Weisfeiler-Lehman Neural Machine) (Zhang and Chen 2017) that improves the overall performance of WLNM reducing the number of hops necessary to properly learn the features. This is especially helpful for smaller graphs, like the housing spatial configurations, whereas the previously mentioned Graph Autoencoder (GAE) requires a bigger network, e.g., the Cora citation dataset (McCallum et al. 2000). In particular, SEAL applies a GNN instead of a fully connected ANN and the gamma-decaying heuristics to increase the performance. In a comparative evaluation, SEAL outperformed GAE (Zhang et al. 2020).

Starting with the original link prediction model, which was originally trained on the Cora dataset, and applying it to our housing graph connections dataset described above, we gradually modified the model improving its performance using the different ANN/GNN options from DGL. This modification technique for models is also known as one of the *transfer learning* methods (Brownlee 2019). Ending up with the GNN configuration consisting of 3 deep layers activated with the *tanh* function and application of *dropout* of 0.5 (instead of 2 layers, *relu activation* and no dropout), the *GraphSAGE* layer structure (Hamilton et al. 2017) and the *dot product calculation* for the final prediction score (see also Figure 4) were kept. The originally used *Adam* optimizer (Kingma and Ba 2014) was replaced by *Nadam* (Dozat 2016) and the number of training cycles could be reduced to 150 from the original 250. The original learning rate of 0.01 was kept. This model configuration provided the best performance (see next sections).

Training, Test and Evaluation

In machine learning-related research, the best practice to automatically find the model with the optimal performance is to train different model configurations on a training dataset, control the training on a test dataset, and then evaluate the models with the best test outcome on the separate validation dataset. One of the common requirements on the validation dataset is the covering of the same distribution of samples as for the training dataset in order to check how well the model has learned the provided samples. This means that the model should be effectively evaluated for use under the real-world conditions.

From the original dataset of 2544 spatial configurations derived from Building Information Modeling (BIM) data, overall, *2075* spatial configurations were used for training, from which *18463* connections for *14261* room entries could be extracted according to the schema described in the *Dataset* section of this paper. From this amount of data, *10%* were used for testing. Additionally, *92* spatial configurations with overall

974 connections for 726 room entries were used to evaluate the model. The rest was filtered out as it did not correspond to the selection criteria described in the *Dataset* section.

The main goal of the evaluation was to check if the GNN model is able to learn all of the latent and explicit semantic features in order to provide a contextually suitable connection prediction. To achieve this, the creation process of the evaluation dataset made sure that all of the features (see Figure 3) were covered by the set of spatial configurations selected for the validation. In order to make sure that the model indeed performs constantly on samples it has never seen before, the spatial configurations for validation were selected randomly during the data pre-processing for the model and the entire process of training-testing-evaluation was repeated multiple times for the best performing model configuration (in this and next sections, only the latest recorded iteration is detailed out).

Results

The results of the training, testing and validation processes revealed that the link prediction model developed using SEAL and DGL is generally able to predict a contextual existence of a connection between two rooms in a spatial configuration for the semantics it has learned from both room and connection entries. In its current state, the training accuracy value converged at approx. *0.93* (see Figure 5), i.e., the model is set to be able to differentiate the existence and non-existence of a connection with precision of *93%*. Even higher is the value of evaluation accuracy: here, an approximate value of 0.98 could be achieved. That is, from the 100% of existing connections selected for evaluation, *98%* were predicted to exist. The loss during the training process converged at the value of approx. *0.39* (see Figure 6). Due to such high accuracy numbers, all results can be considered a success of application of graph neural networks using SEAL and DGL for link prediction in architectural spatial configurations.

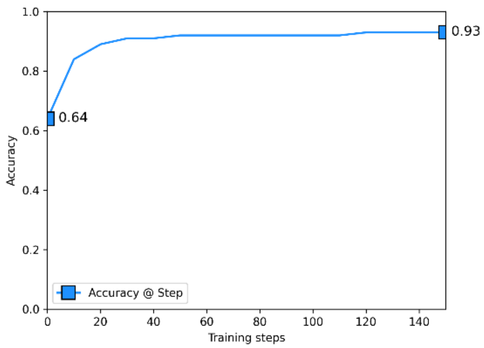


Figure 5
Training accuracy of the link prediction model.

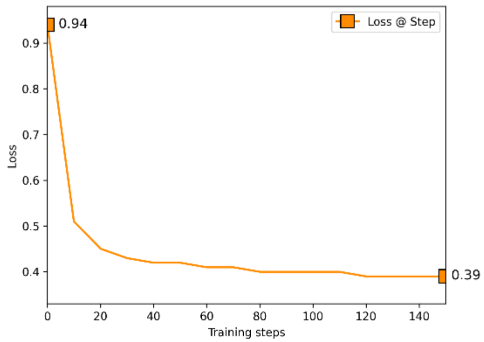


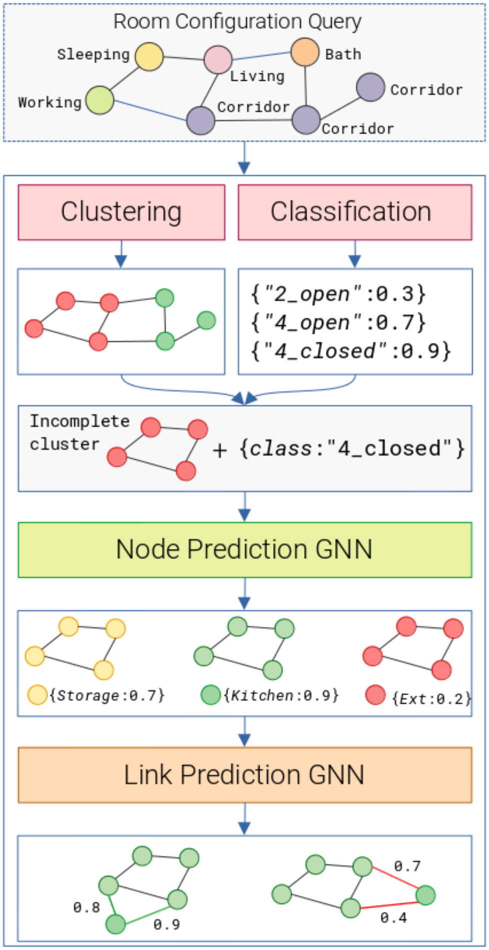
Figure 6
Loss distribution accuracy of the link prediction model.

CONCLUSION AND FUTURE WORK

In this paper, we presented a deep learning model for prediction of connections between rooms in a graph-based spatial configuration. Graph neural networks from the approach SEAL, using the deep learning framework DGL, were applied. The resulting model is able to predict the existence of connections with an accuracy of approx. 93%.

While the link prediction method can be used as a standalone component to identify potential linkages, in the near future it is planned to use it as an integral part of the framework of the research project metis-II for full autocompletion of spatial configurations. In the remainder of this paper, we present this integration, showing which methods complement link prediction in order to get a full autocompletion recommendation.

Figure 7
Integration of link prediction GNN in the generation process of a full autocompletion recommendation.



Integration in the Framework

As depicted in Figure 7, link prediction is the last step of the sequential procedure to generate a full autocompletion recommendation, which consists of the likely missing rooms and the corresponding linkages. The generation of autocompletion starts with the parallel execution of clustering and classification on the current spatial configuration.

During the classification process, the current spatial configuration is first converted into a one-hot encoded tensor representation (see Figure 1) and then classified using a convolutional neural network to find out the housing class of the configuration. The housing class contains the information on the amount of habitable spaces (e.g., 2 or 3) and if the living room is connected with the kitchen via passage (open or closed). An example of such housing class is 3_open.

During the clustering process the current spatial configuration gets divided into different segments (clusters), e.g., using the algorithm (Girvan and Newman 2002). The clusters are then checked for architectural consistency using the consistency checker (Arora et al. 2021) (see also the Dataset section). Inconsistent clusters will be added to the autocompletion query together with the previously determined housing class.

In the next step, a specific node prediction GNN suggests rooms possibly missing in the inconsistent clusters, based on training on the previously recorded consistent clusters. Link prediction GNN then estimates the availability of connections between the suggested and existing rooms, as described in this paper, producing a full autocompletion recommendation. The user is then presented with this recommendation.

Besides the completion of missing entities of a spatial configuration with clustering and link prediction, another autocompletion method will be part of the framework as well. This method is going to apply a pipeline of recurrent neural networks (RNN) to predict the next design step based on the recognized design phase and design intentions. For training of RNNs, quantifiable sketch protocol data generated by architects during floor plan sketching process in the early design process and processed by a specific protocol analyzer tool (Bielski et al. 2022) will be used. A preliminary description of this autocompletion approach was presented in the previous research work (Eisenstadt et al. 2022).

With the autocompletion framework it is intended to provide a tool for architects in the early phases of the design process by complementing the design of architecture with the latest technological developments as proposed by (Fricker et al. 2007) as the optimal collaboration option between the architects and computers.

Further Development

Furthermore, for the upcoming development of the link prediction method it is intended to recommend retroactive changes and so support the architect by drawing attention to potentially flawed connections. This will help provide a consequent insight in order to improve the overall architectural quality of the design.

Source Code

The link prediction model is published on GitHub under an open-source license for use by other researchers and developers, see following URL: https://github.com/metis-caad/link-prediction.

ACKNOWLEDGEMENT

We would like to thank the German Research Foundation (DFG) for funding the *metis* projects.

REFERENCES

Arora, H., Bielski, J., Eisenstadt, V., Langenhan, C., Ziegler, C., Althoff, K.-D. and Dengel, A. (2021). Consistency Checker - An automatic constraint-based evaluator for housing spatial configurations. In Stojakovic, V and Tepavcevic, B (eds.), Towards a new, configurable architecture - *Proceedings of the 39th eCAADe Conference* - Volume 2, University of Novi Sad, Novi Sad, Serbia, 8-10 September 2021, pp. 351-358.

Bielski, J., Langenhan, C., Ziegler, C., Eisenstadt, V., Dengel, A. and Althoff, K.-D. (2022). Quantifying the Intangible, A Tool for Retrospective Protocol Studies of Sketching During the Early Conceptual Design of Architecture. In Jeroen van Ameijde, Nicole Gardner, Kyung Hoon Hyun, Dan Luo, Urvi Sheth (eds.), POST-CARBON - *Proceedings of the 27th CAADRIA Conference*, Sydney, 9-15 April 2022, pp. 403-411.

Brownlee J. (2019) *A Gentle Introduction to Transfer Learning for Deep Learning* [Online]. Available at: https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (Accessed 31 March 2022)

DGL Team (2018) *Link Prediction using Graph Neural Networks* [Online]. Available at: https://docs.dgl.ai/en/0.6.x/tutorials/blitz/4_link_predict.html (Accessed 31 March 2022)

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269-271.

Dozat, T. (2016). Incorporating nesterov momentum into adam. In *Proceedings of the 4th International Conference on Learning Representations* (ICLR 2016).

Eisenstadt, V., Arora, H., Ziegler, C., Bielski, J., Langenhan, C., Althoff, K. D. and Dengel, A. (2021). Exploring optimal ways to represent topological and spatial features of building designs in deep learning methods and applications for architecture. In A. Globa, J. van Ameijde, A. Fingrut, N. Kim, T.T.S. Lo (eds.), PROJECTIONS - *Proceedings of the 26th CAADRIA Conference* - Volume 1, The Chinese University of Hong Kong and Online, Hong Kong, 29 March - 1 April 2021, pp. 191-200.

Eisenstadt, V., Bielski, J., Mete, B., Langenhan, C., Althoff, K.-D. and Dengel, A. (2022). Autocompletion of Floor Plans for the Early Design Phase in Architecture: Foundations, Existing Methods, and Research Outlook. In Jeroen van Ameijde, Nicole Gardner, Kyung Hoon Hyun, Dan Luo, Urvi Sheth (eds.), POST-CARBON - *Proceedings of the 27th CAADRIA Conference*, Sydney, 9-15 April 2022, pp. 323-332.

Fricker, P., Hovestadt, L., Braach, M., Dillenburger, B., Dohmen, P., Rüdenauer, K., Lemmerzahl, S. and Lehnerer, A. (2007). Organised Complexity. Application of Statistical Design in Large-Scale Building Projects. In Predicting the Future [*25th eCAADe Conference Proceedings* / ISBN 978-0-9541183-6-5] Frankfurt am Main (Germany) 26-29 September 2007, pp. 695-701.

Girvan, M., and Newman, M. E. (2002). Community structure in social and biological networks. In *Proceedings of the national academy of sciences*, 99(12), 7821-7826.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N. and Welling, M. (2016) Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Liben-Nowell, D. and Kleinberg, J. (2007) The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7), pp. 1019-1031.

Lutes, J. (2021) *Binning for Feature Engineering in Machine Learning* [Online]. Available at: https://towardsdatascience.com/binning-for-feature-engineering-in-machine-learning-d3b3d76f364a (Accessed 31 March 2022)

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2), pp. 127-163.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

UCL Space Syntax (2022) *Spatial form analysis. How are spatial relations measured and analysed?* [Online]. Available at: https://www.spacesyntax.online/applying-space-syntax/building-methods/spatial-form-analysis/ (Accessed 31 March 2022)

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J. and Zhang, Z. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Zhang, M., and Chen, Y. (2017). Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 575-583.

Zhang, M., and Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31, pp. 5165-5175.

Zhang, M., Li, P., Xia, Y., Wang, K., and Jin, L. (2020). Revisiting graph neural networks for link prediction. *arXiv preprint arXiv:2010.16103*.

Zhou, T., Lü, L., and Zhang, Y. C. (2009). Predicting missing links via local information. *The European Physical Journal B*, 71(4), pp. 623-630.