Comparative Evaluation of Tensor-based Data Representations for Deep Learning Methods in Architecture

Viktor Eisenstadt[1], Hardik Arora[2], Christoph Ziegler[3],
Jessica Bielski[4], Christoph Langenhan[5], Klaus-Dieter Althoff[6],
Andreas Dengel[7]
[1,2,6,7] German Research Center for Artificial Intelligence (DFKI) [3,4,5] Technical University of Munich
[1,2,6,7] {viktor.eisenstadt|hardik.arora|klaus-dieter.althoff|andreas.dengel}@dfki.de [3,4,5] {c.ziegler|j.bielski|langenhan}@tum.de

This paper presents an extended evaluation of tensor-based representations of graph-based architectural room configurations. This experiment is a continuation of examination of recognition of semantic architectural features by contemporary standard deep learning methods. The main aim of this evaluation is to investigate how the deep learning models trained using the relation tensors as data representation means perform on data not available in the training dataset. Using a straightforward classification task, stepwise modifications of the original training dataset and manually created spatial configurations were fed into the models to measure their prediction quality. We hypothesized that the modifications that influence the class label will not decrease this quality, however, this was not confirmed and most likely the latent non-class defining features make up the class for the model. Under specific circumstances, the prediction quality still remained high for the winning relation tensor type.

Keywords: *Deep Learning, Spatial Configuration, Semantic Building Fingerprint*

INTRODUCTION

Deep learning (DL) is widely used for many research areas of computer science, however, its application to computer-aided architectural design (CAAD) is still not as common as in other domains. Mainly this is caused by missing a common representation of spatial configurations that is fully compatible with contemporary DL methods. While images of floor plans can be an obvious choice, they do not explicitly provide semantic information necessary for decision-making of many DL methods, and more suitable ways to represent architectural data in DL exist.

In the context of research on DL-based autocompletion of room configurations, several data representations in the form other than images were investigated (Eisenstadt et al., 2021). As a result, the data structure "relation map" was developed that uses a specific adjacency matrix-based tensor to represent semantic relations between the rooms in a spatial configuration. Using a specifically configured convolutional neural network (CNN), text-based, numerical layer-based, and one-hot vector-based types of relation map were pre-evaluated using the task of classification based on a set of spatial relation classes. This preliminary evaluation was performed automatically by the DL framework, the winning one-hot vector-based relation map type classified the samples with approx. 98% validation accuracy.

In this paper, a more in-depth subsequent evaluation of these representations is presented that investigates how their corresponding deep learning models evaluate spatial configuration samples not present in the training dataset. It is intended to answer the following research questions:

1. Which of the representation type models classifies structurally unknown and manually created samples correctly in the majority of cases?
2. When classifying data initially known but slightly or heavy modified in relational structure, will the predicted class remain appropriate?

Using a set of data augmentation rules to operate on consistently modified datasets for the evaluation and manually design multiple spatial configurations, it is intended to simulate real-world classification usage. The main goal of this evaluation is to find the relation map type that provides the best classification response to changes in the semantic information available in spatial configurations. Additionally, we aim to find out which modifications are responsible for significant changes in the overall classification rate. The results of the evaluation are required for improvement of the CNN-based approach for retrieval of contextually similar floor plans to support early phases of architectural design (Eisenstadt et al., 2020). In this approach, classification of spatial configurations narrows down the set of relevant retrieval candidates based on classes recognized in the query. The winning type of relation map will also be used in a sequential form in the upcoming segmentation-based floor plan autocompletion approach.

RESEARCH CONTEXT & RELATED WORK

During the currently running research project *metis-II* (2020-2023, supported by the DFG - German Research Foundation), we examine and develop DL-based methods and approaches for support of the early conceptual phases in architectural design. Taking into account the vagueness and uncertainty of architectural design data in the form of graph-based spatial configurations, we investigate how auto-completion of floor plans (comparable to e.g., word and sentence completion on keyboards of modern mobile devices) can be achieved using artificial neural networks. Based on early sketches of the building designs, rooms and the possible relations between them are suggested to the architect to enhance the early ideation process. The auto-completion methods are intended to be a helpful tool for architects during the early conceptual process supporting them in working more efficiently, while being able to focus on the creative aspects/tasks. Providing the architects with the different design continuation options, it is intended to create interaction patterns to assess their own design decisions and explore the possible further developments of the current spatial configuration state.

To represent the spatial configuration in the form compatible with the deep learning methods, but also keep the relevant semantic information entered by the architect, we use a number of *Semantic Building Fingerprint* graphs (SBF), where nodes represent rooms and edges represent relations (connections) between the rooms. The information encoded as an SBF-based graph contains topological as well as non-topological data distributed among several different patterns (see figure 1). For use of these patterns in DL methods, they are transformed into the numerical tensor format "relation map" which represents a modified adjacency matrix of the spatial configuration graph using relations between the rooms of the graph to encode semantic information. The re-

Figure 1
Examples of semantic building fingerprints (SBF). With topology (1-4): semantic spatial graph, through-path-graph, semantic spatial connection graph, spatial distance. Without topology (5, 6): envelope area, center of room.

lation map concept was designed using the idea of reverse compatibility, i.e., the tensor can be transformed back into spatial configuration and displayed as such in the user interface. The early types of relation maps were used for data augmentation (Arora et al., 2020). The further developments of them were preliminary evaluated (see next section), but until now not applied in any approach as they were not extensively tested yet for unexpected classification cases, this testing will be described in this paper.

Currently, deep learning approaches in architecture use raster- or vector-based images of floor plans, mostly due to their availability and as an obvious choice for use in popular DL methods, such as CNNs, for which many applicable and extendable applications examples from other domains already exist. However, as examined during development of approaches for our research project, such entirely pixel- or vector-based color data does not provide the necessary semantic information. For example, for semantic building fingerprint-based auto-completion, the recognition of relational dependencies between the rooms is required that is not possible with image-based data. Nevertheless, successful approaches for purposes other than ours use architectural image data: as examples, search for similar designs (Sharma et al., 2017), modification of the design style (Newton, 2019; Silvestre et al., 2016), or estimation of the layout in 3D (Sun et al., 2019) can be named.

An additional problem, that became evident during the research project, is that even when non-image-based data samples (e.g., as graphs in XML format) are available, a suitable common representation of spatial configurations for DL is missing, as the sufficient amount of data might be available from different sources that do not have common semantic representation requirements and the pure XML-based data is not a numerical tensor format required for the contemporary DL frameworks e.g. TensorFlow or PyTorch. While it is also possible to use specific DL abstraction layers, such as *Deep Graph Library* (Wang et al., 2019), they do not provide architecture-specific DL representations and/or models and they have to be additionally developed nevertheless.

DL REPRESENTATIONS & PRE-EVALUATION

To overcome the deep learning representation problems described above, a number of architecture-specific tensor-based data structures were developed that can be used with standard methods of contemporary DL frameworks. All these representations are based on the common concept of "relation map", an adjacency matrix modified to encode the relevant semantic features detected in the topology of the spatial configuration graph. Each row of such matrix tensors stands for a particular room of the floor plan and its outgoing connections to other rooms. Geometrical parameters of the building design are not considered. Relation maps are inspired by architectural morphospaces (Steadman and Mitchell 2010) and geometry maps (De Miguel et al., 2019), but also by Hickman and Krolik (2009) who used the adjacency matrices to encode general availability of connections in spatial configurations.

The essential semantic entity of a relation map is the *relation code*, that numerically encodes the connection information in a triple **<source room, target room, connection type>** in the cells of the matrix rows using a specific typology in which each room and connection type are assigned to the specific integer numerical value (see Figure 2). For example, in the relation code **562**, the source room type *Living*

(5) is connected to the target room type *Bathroom* (6) by the connection type *Door* (2). Currently, five types of the relation map exist. The *simple relation map* is a 1D tensor and uses relation codes in the form described above. The *zoned relation map* enriches the relation codes with the information about *architectural room zones*, the categorical taxonomy of room types that groups them by their functionality (Langenhan, 2017). For example, the relation code **35162** represents the room type *Kitchen* (3) from the *Habitation* zone (5) connected to the room type *Corridor* (1) from the *Service* zone (6) by a *Door* (2). The zoned connection map was developed to eliminate possible repetition of relation codes. Later, more advanced types *multilayer map*, *one-hot encoded map*, and *textual map* were developed.

The multilayer map (see Figure 2, upper right part) was developed for the situations where the typology of room and connection types is not explicitly available for the given dataset of spatial layouts. To encode relevant semantic information in samples from such datasets, the conversion methods of the multilayer map use the cryptology method of *hashing* of values detected as possible room and relation types. Provided that the data was unified during the creation of the dataset, the hashing will return the same values for the same room and connection types. The subsequent encoding into decimal fraction will result in the 3-element relation code array for source room, target room, and connection type. For example, the connection *Living <- Door -> Bathroom* can be encoded as an array **[0.15383, 0.25237, 0.32474]**. Inspired by the RGB scheme channels, a multi-layered 3D map can then be constructed consisting of the source room layer, target room layer, and the connection type layer. Each layer will use the values from the corresponding array index only.

The one-hot encoded map (see Figure 2, middle part) was designed to properly encode categorical data instead of categorization of pure numerical data, which is used in all previously presented maps. Using the classic one-hot encoding technique for categorical data, the room and connection types are encoded as *one-hot vectors*, the position of **1** in the vector defines the corresponding type. All other positions are filled with **0**. Each vector has as many elements as the overall number of types for the respective information type (i.e., room or connection type). This method avoids summarization of semantic information in the form of source, target, and connection into one numerical value, where the order of the numbers and the position further left or further right in the relation code has a larger or smaller influence on this value. Additionally, the one-hot representation eliminates the risk of assigning categories to numbers which can be interpreted as ranking and/or similarity measure. For example, using the typology numbers of Figure 2, the difference between *Toilet* (7) and *Sleeping* (2) would be greater than between *Living* (5) and *Corridor* (1): **7-2=5** vs. **5-1=4**.

Finally, the textual relation map (see Figure 2, lower part) was developed for adaptation to the changes in the number of room types or edge types in the typology preserving the original shape of the tensor and to solve the problem of sparsity (for example, for spatial configurations with a small number of connections in relation to the number of rooms). The textual relation map is a variation of the simple relation map, that uses sentences to represent connections between rooms as relation codes. An example of such a sentence is **'living connects with sleeping using wall'**. To represent non-existing connections between two rooms the textual placeholder **'no connection'** is used (instead of using 0 like the number- or vector-based maps described above). Given that this map representation is text-based, it can also be used in a different form for multiple DL approaches including sequence learning or text classification.

For the last three representations described above, a preliminary evaluation (Eisenstadt et al., 2021) was performed, that should provide an initial answer on the question if the relation map tensors are able to properly encode the relevant semantic information contained in the spatial configuration graphs, so that the DL models can understand this information and meet the correct deci-

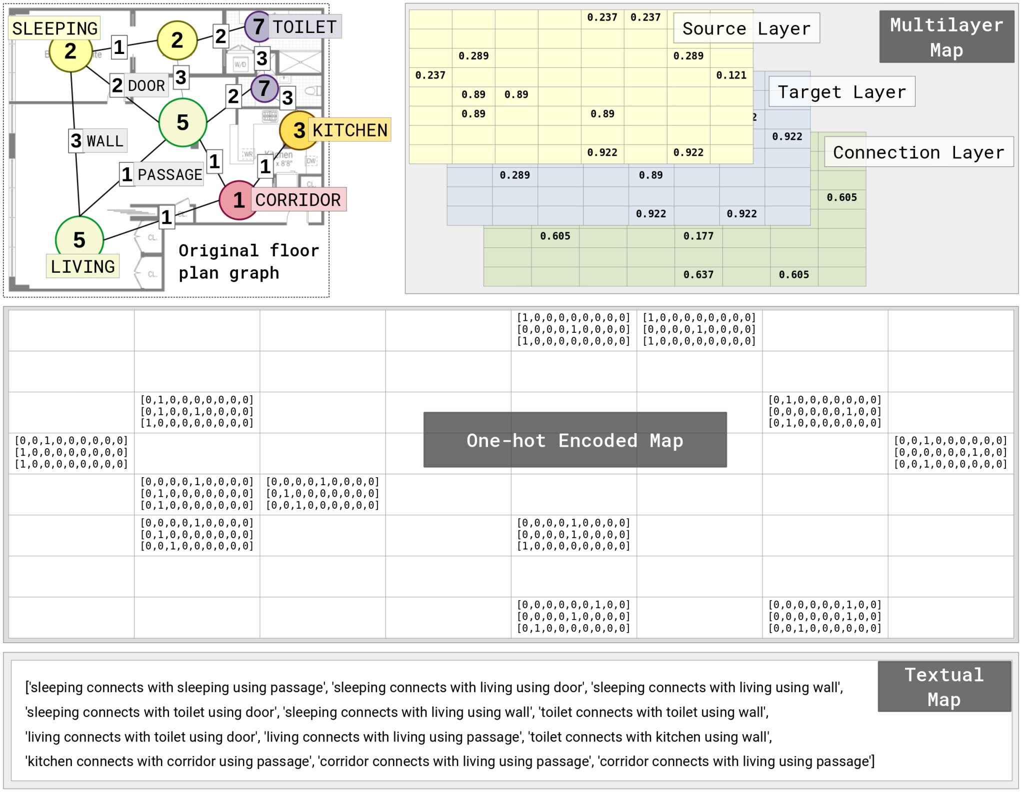


Figure 2
Original floor plan SBF and its conversions into multilayer map, one-hot encoded map, and textual map. For selected numerical encodings of the underlying typology, their corresponding room and connection types are shown.

sion, e.g., correctly classify the spatial configuration. An automated experiment using the straightforward classification process was selected as the evaluation method due to the availability of classification and its validation techniques directly in the modern DL frameworks. For training the DL models in the form of CNNs, a dataset of 2544 housing floor plans was used. It was based on the original 200 data samples that were extended using a number of variation and consistency rules. The measurement of training and validation accuracy was performed automatically using the built-in DL framework tools. The results of this pre-evaluation indicated that the DL models are able to process and understand the latent semantic structure of the selected relation map types and the information contained in them. A max. validation accuracy of 98% was achieved for the winning type, the one-hot encoded map. The multilayer map and textual map could achieve a max. of 84% and 83% respectively. *These results are our current benchmarks for validation of the prediction quality of the models.*

While the pre-evaluation provided satisfactory results that confirmed the direction of our research efforts, it was still not clear how the DL models trained on the selected relation map types would react when introduced spatial configurations not available in the training dataset. To answer this question, an extended evaluation that involved again the multilayer, one-hot encoded, and textual map was conducted and will be described in the next section.

EXTENDED COMPARATIVE EVALUATION

The extended comparative evaluation of tensor-based representations of semantic building information was performed to finalize the initial exploration of their suitability for standard DL tasks. While in the pre-evaluation described above an initial hypothesis on suitability should be tested using no means other than the built-in tools of DL frameworks, in this evaluation, the selected relation maps should prove their potential for use in DL-based applications developed in context of the research project. Identically to the pre-evaluation, tests shall show which relation map type provides the best performance.

To answer the question described above, a *double-phase evaluation scenario* was developed which was aimed at extensive testing of class prediction quality of the relation map models. The tests were conducted under different *grades of modification* of the original training dataset and spatial configuration samples not available in the original dataset, but *created manually* using the same format and consistency checking tools. The relation map DL models from the preliminary evaluation were reused without re-training them. The class labels were reused from the pre-evaluation as well: each class indicated the *amount of habitable spaces*, i.e., Sleeping, Living, Children, Working, and Room (generic room label), 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 class label is **"3_closed"**. Overall **10** classes were created.

The background idea of the evaluation was to perform the classification of the manually created spatial configurations first, in order to get an immediate initial estimation on how good the models can classify unknown cases. Based on the result of this initial estimation, i.e. low or high prediction quality, the task would be to investigate what modifications might have caused it and if this result is a random occurrence. That is, either the most likely failure trail should be reconstructed (in case of low quality) and/or the possible fluctuations in the classification rate should be tracked (for both low and high rates).

In order to measure the sufficiency of the classification of unknown and modified samples by the models, it was decided to calculate the percentage of correctly predicted labels using following two specific metrics and then compare them to the benchmarks from the pre-evaluation (see previous section):

- **First result** out of overall 10 possible class labels outputs in order to follow the classic classification accuracy measurement
- **3 first results** in order to assess if the sufficient coverage of correct classification was available nevertheless

The reason to introduce the latter metrics was an existing research application use case. The semantic building graph retrieval method developed for the research project uses a max. of 3 class labels to make up a set of contextually suitable retrieval candidates for the subsequent graph isomorphism process.

In the next sections, the evaluation of manually created cases and the subsequent investigation phase using modified spatial configuration data samples will be described. For each of them, the corresponding measurement of prediction quality using the two metrics described above will be presented.

Phase 1: Evaluation of Manual Graphs

As explained above, the evaluation of manually created spatial configuration graphs should provide an immediate estimation of classification quality of the convolutional neural networks trained using one-hot encoded, multilayer, and textual maps on the same dataset and with the same model configuration. Overall, **12** manual room graphs based on a selection of already existing buildings were created using a specific spatial configurations editing tool *RoomConf Editor* (source code available at: *github.com/cenetp/roomconf-editor*) that allows for creation of graphs in the format *AGraphML*, which provides compatibility for conversion into relation maps. The converters are published under *github.com/metis-caad/roomconf-converter*.

Table 1
Results of prediction quality in % (higher = better) on manually created spatial configuration graphs.

Map type	*First result*	*All 3 results*
One-hot encoded	41.67%	91.67%
Multilayer	16.67%	66.67%
Textual	8.33%	50.0%

These 12 manually created spatial configuration graphs were then converted as well and fed as relation map tensor queries into the corresponding convolutional neural networks of the evaluated map types. Table 1 shows the measured prediction quality in the form of classification rates achieved by each map type in this evaluation phase.

The majority of the results *did not provide a sufficiently high percentage of correct predictions* when comparing them to the benchmarks from the pre-evaluation. Considering the first result only (which was the training and validation accuracy metrics in the pre-evaluation) none of the map models performed as good as in the pre-evaluation. Except for the performance of the one-hot encoded map, other maps did not provide a good performance on the first 3 results as well. That is, in the subsequent investigation phase, it should be examined what modifications might lead to the decrease of prediction quality and if the one-hot encoded map's result on the 3 first samples is not just a randomly achieved high rate.

Table 2
Amounts of data samples for each evaluation step for each evaluation path of the investigation phase.

⇣ **Step / Path** ⇢	*SEPARATED*	*SUBSUMED*
1	1672	1672
2 (1+2 Subs.)	3406	3377
3 (1+2+3 Subs.)	2544	4894
4 (1+2+3+4 Subs.)	2057	2356

Phase 2: Investigative Evaluation

Two investigative evaluation paths, *SUBSUMED* and *SEPARATED* (see Figure 3), were introduced to examine findings of the manual graph evaluation. Both paths used a set of **4 dataset modification steps**.

The first modification step was identical for both investigation paths, however, the further application of modifications differed between them:

1. *SUBSUMED*: each subsequent modification step after the first step is performed on the already modified dataset from the previous step
2. *SEPARATED*: each modification step is performed on the original training dataset

All modification steps were defined by the architecture domain experts from the research project *metis-II*. After each modification process performed by a specific *rule-based data augmentation* tool, the new modified dataset was *checked for consistency* by a specific tool developed according to the rules defined by the domain experts during the pre-evaluation. The modified graphs that did not adhere to the consistency rules were left out from the classification process. Following Listings show the modification rules for all 4 steps of the evaluation paths. Table 2 shows the amounts of samples remained for each step after modification and consistency check.

```
Step 1: Modification of access type
  Replace ENTRANCE with DOOR
  IF no ENTRANCE available THEN
   ↪ PASSAGE and DOOR can be
   ↪ interchanged between CORRIDOR
   ↪ and LIVING
  IF CORRIDOR and LIVING are not
   ↪ connected THEN replace random
   ↪ PASSAGE with DOOR (or other way
   ↪ round) except the PASSAGE
   ↪ between KITCHEN and LIVING

Step 2: Replacement of habitable and
 ↪ non-habitable spaces
  The room types ROOM and CHILDREN /
   ↪ WORKING can be interchanged
  IF no ROOM available, THEN one of
   ↪ the following rules applies:
```

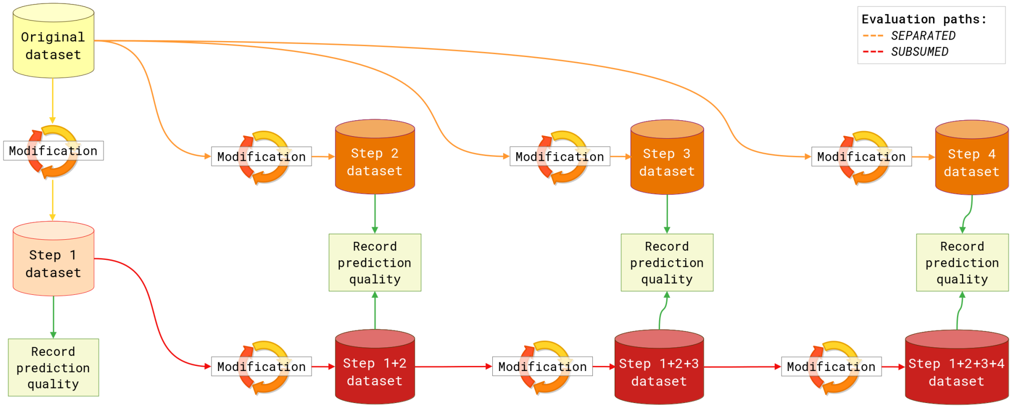


Figure 3
Overview of the investigation phase of the extended comparative evaluation.

```
    BATH or STORAGE are
     ↪ interchangeable with TOILET
     ↪ but not with each other
    CHILDREN and WORKING are
     ↪ interchangeable

Step 3: Adding / removing of non-
 ↪ habitable rooms to keep the
 ↪ current labels intact
  Add one from {CORRIDOR, STORAGE,
   ↪ KITCHEN, TOILET, BATH}; this
   ↪ will not change the amount of
   ↪ habitable rooms
  Randomly, one of {TOILET, KITCHEN,
   ↪ BATH} can be added and connected
   ↪  to CORRIDOR

Step 4: Breaking the label (re-
 ↪ labeling) by adding or removing
 ↪ habitable rooms
  Add one from {ROOM, LIVING, SLEEPING
   ↪ , WORKING, CHILDREN}; this will
   ↪ change the amount of habitable
   ↪ rooms and break the label
  IF there is CORRIDOR THEN add a new
   ↪ ROOM to it with DOOR or PASSAGE
  Randomly, one of {CHILDREN, ROOM,
   ↪ WORKING} can be removed
```

As can be read from the list of modification steps, the modification rules were defined with a specific goal each, increasing the modification strength from step to step. The idea behind this order was the intention to detect when it gets or starts to get better or worse, to identify “good” or “bad” intermediate modifications. The main research question was to find out what provided a bigger influence? “Hard facts” that define the class label from the architectural point of view (openness/closedness, number of habitable spaces) or non-class-defining latent features? Maybe the hard facts are in fact not what defines the class for the DL model? Following two hypotheses were defined for the investigative evaluation:

- *Hypothesis 1:* Prediction quality does not decrease if class-defining features are changed and another class is the result of this modification (Step 4)
- *Hypothesis 2:* Prediction quality should not heavily decrease if “latent features” are changed (Steps 1-3)

Results

The results of the investigative evaluation (see Figures 4 and 5) revealed a general pattern that all maps react similarly to the changes to the data available in the dataset they were trained on. *The stronger the modification the lower is the rate of correctly predicted classes.* The highest influence on the classification

Figure 4
Results from the SEPARATED evaluation path. M stands for the prediction quality on manual graphs, 'all' stands for 'All 3 results' (see Table 1).

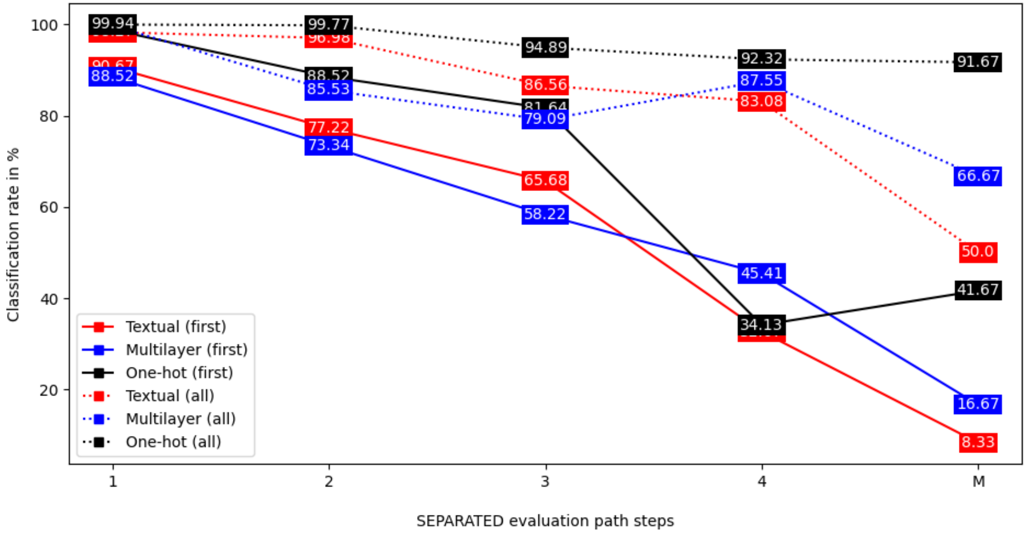
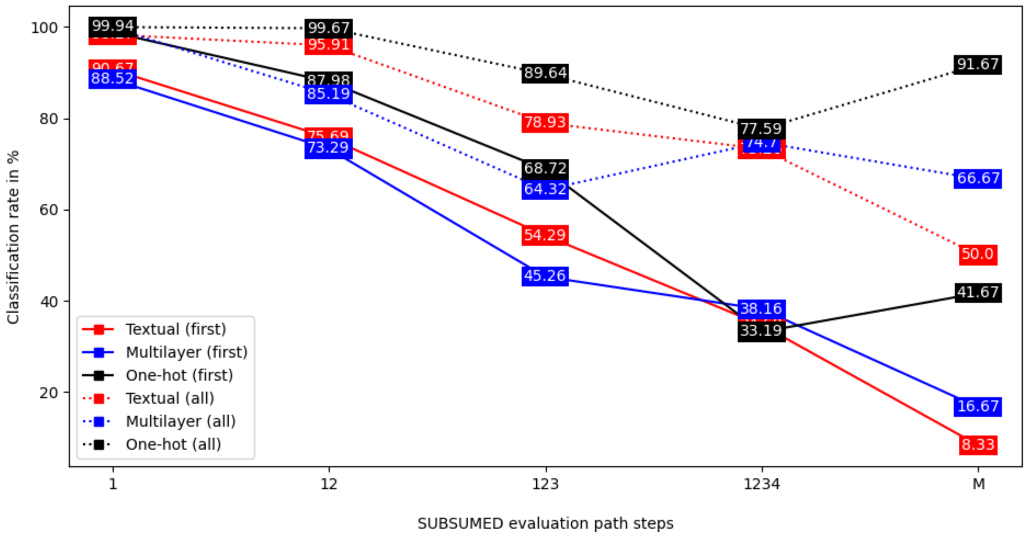


Figure 5
Results from the SUBSUMED evaluation path. Steps 12, 123, and 1234 indicate modifications subsumed from the previous modification step(s).



rate has the *breaking of the class label*, which *does not confirm Hypothesis 1*. The CNNs seem to take the latent features into account as well, and most likely even *build their classification decision* on said features and *not on the label-defining* ones.

Moreover, examining the *SUBSUMED* path, it can be concluded that the *application of further latent modifications on already modified data has a stronger and more stable influence on prediction quality*, resulting in mostly stable decrease in prediction quality. For the *SEPARATED* path, this behavior could be detected only partly, as the first 3 steps provide somewhat stable prediction rates, with a strong decline in steps 4 and M (manual graphs). This *partly confirms Hypothesis 2*, as at least for *SEPARATED* path heavy and sudden decreases could not be recorded.

As an exception from the conclusions described above, the performance of one-hot encoded map can be seen. It increased its rate for the manually created samples in the majority of cases. While the number of 12 cannot be seen as a significant comparison number, a *clear tendency towards the stable performance of one-hot encoded map* can be seen when considering all 3 first results. This is a clear answer that the high classification rate of manual graphs for this type of relation map is not random, and the selection of 3 class labels most likely guarantees that the correct label is available among the first 3 results. This can be seen as improvement for the floor plan retrieval application, as the determination of the suitable search context will be more precise based on the results achieved by the investigative evaluation.

CONCLUSION & FUTURE WORK

In this paper, we presented an extended comparative evaluation of deep learning models that use the tensor-based data structure "relation map" for semantic building fingerprint data. Using a classification task, three types of the relation map were evaluated to find out how the map-based DL models react to data not available in the training dataset. The results revealed that the stronger modified data influences the decrease in prediction quality, except for the winning type, the one-hot encoded relation map. Investigating which modifications can be responsible for decline in prediction rate, we came to the conclusion that modification of features that influence the class label is the most likely candidate. For the future, a user study is planned, where the DL models should justify and make transparent their classification decision using different explainable AI methods.

REFERENCES

Arora, H, Langenhan, C, Petzold, F, Eisenstadt, V and Althoff, KD 2020 'METIS-GAN: An approach to generate spatial configurations using deep learning and semantic building models', *ECPPM 2020-2021*

Eisenstadt, V, Arora, H, Ziegler, C, Bielski, J, Langenhan, C, Althoff, KD and Dengel, A 2021 'Exploring optimal ways to represent topological and spatial features of building designs in deep learning methods and applications for architecture', *CAADRIA 2021*

Eisenstadt, V, Langenhan, C, Althoff, KD and Dengel, A 2020 'Improved and Visually Enhanced Case-Based Retrieval of Room Configurations for Assistance in Architectural Design Education', *ICCBR 2020*

Hickman, G. and Krolik, J.L. 2009, 'A graph-theoretic approach to constrained floor plan estimation from radar measurements', *IEEE transactions on signal processing*, 57(5), pp. 1877-1888

Langenhan, C 2017 'Datenmanagement in der Architektur', *TU Munich Doctoral Diss.*

de Miguel, J 2019 'Deep Form Finding-Using Variational Autoencoders for deep form finding of structural typologies', *eCAADeSIGraDi 2019*

Newton, D 2019 'Deep Generative Learning for the Generation and Analysis of Architectural Plans with Small Datasets', *eCAADeSIGraDi 2019*

Sharma, D, Gupta, N, Chattopadhyay, C and Mehta, S 2017 'DANIEL: A deep architecture for automatic analysis and retrieval of building floor plans', *14th IAPR ICDAR*, pp. 420-425

Silvestre, J and Ikeda, Y 2016 'Artificial imagination of architecture with deep convolutional neural network', *CAADRIA 2016*

Steadman, P and Mitchell, LJ 2010, 'Architectural morphospace: mapping worlds of built forms', *Environment and Planning B: Planning and Design*, 37(2), pp. 197-220

Sun, C, Hsiao, CW, Sun, M and Chen, HT 2019 'Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation', *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1047-1056

Wang, M, Yu, L, Zheng, D, Gan, Q, Gai, Y and Ye, Z 2019, 'Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs', *arXiv:1909.01315*