International Cartographic Association
Association Cartographique Internationale

CNN-Based Geometric Feature Embedding Using Coordinates for Cartographic Generalization Tasks on Building Footprints

Matthias Wamhoff [a]*, Julien Baerenzung [a], Lilli Kaufhold [a], Martin Kada [a]

[a] *Institute of Geodesy and Geoinformation Science, Technische Universität Berlin, 10553 Berlin, Germany, matthias.wamhoff@tu-berlin.de, baerenzung@tu-berlin.de, lilli.kaufhold@tu-berlin.de, martin.kada@tu-berlin.de*

* Corresponding author

Keywords: GeoAI, Cartographic Generalization, Self-Supervised Learning, Building Simplification, Neural Network

Abstract: Cartographic Generalization is a fundamental process in automatic map generation that mostly relies on rule-based constraints. Previous methods that aim at learning this generalization process from data either feed rasterized grids into Convolutional Neural Networks (CNN) or compute hand-designed features to process vector data with Graph Neural Networks (GNN's). However, there is little work that investigates the application of Deep Learning models directly on the vector coordinates. In this work we demonstrate the efficacy of CNN based architectures to highly constrained geometrical spaces such as building footprints, eliminating the need for more complex architectures. To remedy the lack of annotated data we propose to train geometrical feature embeddings in a self-supervised fashion to directly approximate geometrical properties of local triangles in the building footprints, rather than manually engineering Geometrical Features. We validate our approach on a Building Classification and cartographic generalisation task, outperforming previous methods.

1. Introduction

Cartographic Generalisation is a fundamental process in visualizing digital maps across multiple scales while preserving essential geographic characteristics. Within this domain, Building Simplification represents a particularly challenging task that traditionally relies on rule-based approaches, requiring significant domain expertise and manual work. On the other hand, Deep Learning methodologies offer promising avenues for automating this process, by extracting statistical patterns entirely from the data. Thus, they offer a much more flexible way of describing the abundant variations seen in cartographic generalisation.

Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image processing, making them attractive for cartographic tasks. However, rasterizing vector-based geographical data to apply these techniques introduces artifacts that compromise geometric integrity. The alternative approach of applying Deep Learning directly to vector data faces its own significant challenges. Firstly, CNNs seem to work particularly well in high dimensions, from which lower dimensional intrinsic representations can be extracted, but struggle on already naturally lower dimensional data (Pope et al., 2021). Furthermore, they fundamentally rely on the inductive bias of regularly sampled grids, where convolution maintains a linear shift invariant system only when the underlying continuous signal is sampled uniformly. This translation invariance, crucial to CNNs' success in image domains, breaks down when applied to irregularly structured vector geometries.

Recent advancements, summarized under the term "Geometric Deep Learning" (Bronstein et al., 2017), aim at closing this gap. Out of these efforts, most notably Graph Neural Networks (GNN) emerged that can process vector data (Wu et al., 2020). Albeit presenting an interesting candidate for Building Simplification, we argue that GNN's introduce unnecessary complexity and are not necessarily optimal for the well-structured nature of building footprints. These polygonal representations exhibit predictable topological relationships, where each vertex connects to exactly two neighbors in a fixed sequence. This constrained structure allows us to employ simpler 1D convolutions with a filter size of 3, efficiently aggregating information from immediate neighboring vertices while maintaining computational efficiency. For instance, the graph convolutional operator (GCN) from Kipf and Welling (2016) can be written as

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{l}W^{l}), \tag{1}$$

where $\tilde{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix with self-loops, $\tilde{D} \in \mathbb{R}^{n\times n}$ is the degree matrix, σ is a non-linearity, $H^{l} \in \mathbb{R}^{n\times d}$ and $H^{l+1} \in \mathbb{R}^{n\times k}$ are the $l-th$ and $l+1$-th feature map, respectively and $W^{l} \in \mathbb{R}^{n\times k}$ is the weight matrix of layer l. $H^{l}W$ linearly transforms the feature activations through a learnable convolutional filter of size 1. The result is aggregated across neighbouring nodes by multiplying with the adjacency matrix and normalized by division with the node degree. One should note that this normalization becomes unnecessary in graphs with fixed node degree. Conversely, a standard convolutional layer with filter size 3 and circular padding can be seen to equally implement message passing on a graph, but with more flexibility, as it allows to individually weight the contributions across neighboring nodes and different feature channels (See Figure 1). Crucially, this only holds for graphs with a fixed node degree of 2 and a well-defined neighborhood relation.

Existing Deep Learning based methods on building polygons typically augment raw coordinate data with handcrafted features, such as differences to neighboring vertices, rotation angles, and edge lengths, to project each node into a local, translation invariant coordinate system.

This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

While effective, this approach still relies on human expertise to design appropriate feature transformations.

Instead of engineering features that have certain desirable properties, Geometric Deep Learning frameworks aim at defining novel model architectures that can directly process unstructured vector data and incorporate certain geometrical knowledge by design. Many methods have focused on defining a novel convolution operator that abstracts from the original grid assumption towards a definition on unordered vector geometries. In this formulation, a function with trainable parameters that directly takes vector coordinates as input replaces the weights in standard convolutional operators. Thus, these novel convolutional layers are defined over a continuous vector space, rather than a discrete grid. Thereby rendering learned feature representations translation invariant even for irregularly structured inputs (Simonovsky and Komodakis, 2017, Wu et al., 2019, Groh et al., 2018, Wang et al., 2018, Fey et al., 2018).

In contrast, we propose an end-to-end learning paradigm that directly processes vector coordinates, without the need to manually engineer Geometric Features. Instead of relying on a more complex model, we introduce a relatively simple CNN based Feature Extraction Block and assess its ability to approximate geometrical parameters of local triangles in building footprints. A major advantage of this self-supervised regression task is that it does not require any manually labeled data. By stacking several of these modules, we construct deeper models that can fully realize the potential of self-supervised pretraining. The Geometric Embeddings learned in this self-supervised task then serve as initialization for more complex tasks. We validate our approach on a Building Classification (Liu et al., 2021) and a Building Simplification problem proposed by Zhou et al. (2023). For the Simplification task, we propose an Encoder-Decoder structure, akin to Unet (Ronneberger et al., 2015), which is based on our Feature Extraction Block.

2. Related Work

Vector based representations of buildings offer advantages for spatial analysis over rasterized data, yet direct processing of vector coordinates in Deep Learning for building data remains largely underexplored, due to the challenges presented by its irregularly sampled and lower dimensional nature.

Liu et al. (2021) propose Deep Point Convolutional Networks (DPCN), replacing typical convolutional layers by subtracting neighboring node activations. They stack these blocks into a modular architecture similar to Pointnet (Qi et al., 2017) for Building Classification. They also resample footprint vertices equidistantly to a fixed number on the polygon's boundary, restoring the inductive bias for CNNs that typically excel on regularly sampled grids.

Mai et al. (2023) compute differences to the p-next neighbors and feed these enhanced feature representations into a ResNet (He et al., 2016). Furthermore, they make a principled analysis of the invariances that a polygon embedding should exhibit. These are loop origin, trivial vertex and part permutation invariance, as well as topological awareness. Out of these four, they achieve loop-origin invariance with ResNet by applying circular padding.

Zhou et al. (2023) feed edge lengths, rotation angle, and edge direction into a GNN in a Building Simplification task. GNNs seem to be a promising alternative for polygonal data, as they can incorporate the inductive biases inherent in graph structures. However, we argue that GNNs are not necessarily optimal for data with highly constrained geometric properties, like building footprints, where vertices have a fixed degree of two.

Veer et al. (2018) apply a relatively shallow CNN directly on vector coordinates, but use data binning and zero padding to remedy the variable sized data of building footprints. In our experiments we discovered a stark degradation of performance when using binning methods and we resort to a batch size of 1 instead.

Unlike previous methods, we design a CNN based Feature Extraction Module that directly processes vector coordinates. Instead of relying on handcrafted features or complex architectures to incorporate geometrical knowledge into the model, we develop a self-supervised pretraining scheme that enables the model to extract relevant Geometric Embeddings. In the following we will describe our method in more detail. Next, will stack this Feature Extraction module several times to solve the more complex task of Building Shape Classification, before we build an even deeper Unet-like architecture from the same module and turn to the Building Simplification task from Zhou et al. (2023). Finally, we conclude our presented work and discuss potential caveats and future research directions.

3. Methodology and Experiments

In this section we describe our CNN based Feature Extraction module in detail and show that it can extract geometrically relevant features on a self-supervised regression task. Building on this, we turn to more complex Building Classification and Simplification tasks and describe the particular architecture for each task in detail. Furthermore, we directly report results on each subtask.

We propose a CNN based geometrical Feature Extraction module that uses 1D convolutions with filter size 3 and circular padding as a way to aggregate features from neighboring vertices. An illustration of this can be seen in Figure 1. As described in (Mai et al., 2023), circular padding adds the last element in the node sequence to the front and the first element to the end, so that the convolution result becomes invariant to the choice of start node of the sequence. The aggregated features are then refined through a 3-layer node-wise MLP, implemented with 1D convolutions which are followed by a normalization layer and ReLU activation (Figure 2). Notably, convolutions inherently handle variable-sized input, making this approach naturally compatible with building footprints of different sizes. In order to be able to process geometries with batches, one needs to first divide the data into buckets with same length (Veer et al., 2018). However, we have observed a severe decline in

Proceedings of the International Cartographic Association, 7, 26, 2025 | https://doi.org/10.5194/ica-proc-7-26-2025
32nd International Cartographic Conference (ICC 2025), 17–22 August 2025, Vancouver, Canada.
This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

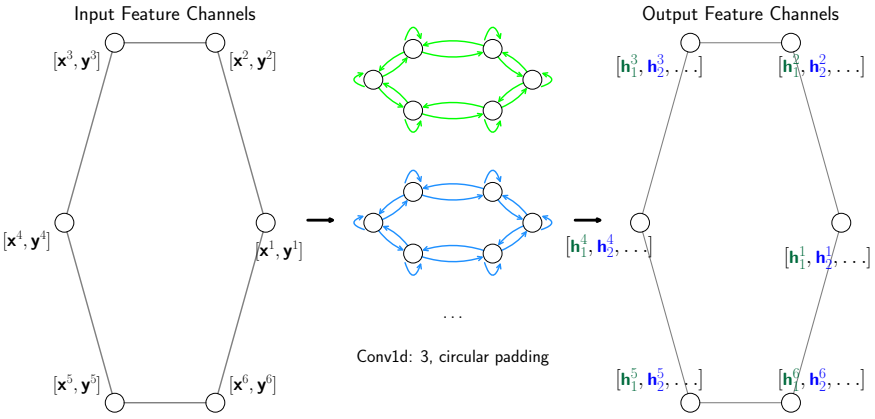

Figure 1. Visualization of a standard convolutional layer with filter size 3 and circular padding for local feature aggregation. The Input features of the first layer are the (x, y) coordinates of the footprint vertices

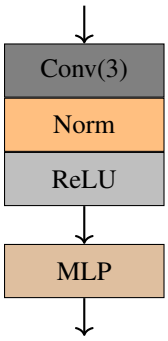

Figure 2. Architecture of the proposed building block block. Conv(3) is a 1d convolution with filter size 3 and circular padding. The MLP are 3 linear layers, implemented as convolutions of size 1, followed by normalization and ReLU.

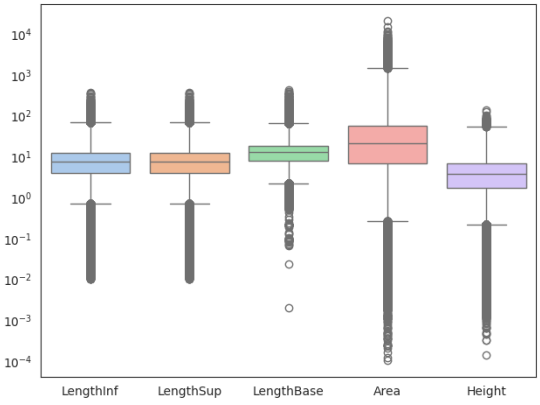

Figure 3. Box plots of geometric properties from local triangles in our training set, comprising $N = 30352$ building footprints. The sine and cosine are omitted. All values are plotted in log scale. Outliers are plotted as dots outside of the delimiters, while the colored area inside the box shows the first quartile.

performance for these bucketing techniques and resort to a batch size of 1, instead.

We normalize the data by subtracting footprint centroids from each coordinate. Generally, we do not see any improvements from normalizing coordinates by dividing with the standard deviation as in (Veer et al., 2018). In all of our experiments, we use Adam optimization with weight decay of 1e-6 (Kingma, 2014), Cosine Annealing to adapt the learning rate (Loshchilov and Hutter, 2016) and ReLU activation functions.

Firstly, we show that we can extract geometrical properties of local triangles using this simplistic building block. We will later leverage this self-supervised pretraining scheme to effectively tune a deeper model for Building Classification and Simplification.

3.1 Self-Supervised Local Feature Extraction

Building on the tradition of Geometric Feature Engineering in vector data processing, we propose a novel approach that leverages Deep Learning to discover geometrically relevant features rather than manually crafting them. Our methods learns to extract meaningful geometric information directly from vector coordinates through self training, effectively learning embeddings that capture the spatial relationships inherent in building footprints.

In a preliminary task, we aim to regress geometrical properties of local triangles within the building footprints. To this end, we employ the shallow building block described before, with one convolutional layer and kernel size 3, followed by a 3-layer MLP. Each layer has 256 output channels. Finally, a linear convolution with kernel size 1 produces the regression targets. We omit any normalization layers due to the shallowness of the model and the simplistic nature of the task.

We use the footprint dataset from Zhou et al. (2023), Feng et al. (2019) in the scale 1:10.000. Any footprints with a sequence length greater than 40 and with 3 consecutive collinear vertices are discarded, arriving at a total dataset

Proceedings of the International Cartographic Association, 7, 26, 2025 | https://doi.org/10.5194/ica-proc-7-26-2025
32nd International Cartographic Conference (ICC 2025), 17–22 August 2025, Vancouver, Canada.
This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

size of $N = 37941$ buildings. We split these buildings into train-, valid- and test set with a ratio of $8:1:1$, respectively. We then calculate $k = 7$ geometrical properties from local triangles within the building footprints, namely edge lengths of the triangle legs and base, sine+1, cosine+1, area and height of the triangle. The model is trained with the mean absolute error (MAE) loss for 200 Epochs. The learning rate is gradually annealed between λ_0 =1e-4 and λ_{min} =1e-6. Since the individual distributions of the respective regression targets vary rather significantly, we divide the loss by the standard deviation for each target

$$L = \sum_{i}^{k} \sum_{j}^{n} \frac{1}{n\sigma_i} |\hat{y}_i^j - y_i^j|. \tag{2}$$

Here, $k = 7$ is the number of regression targets and n is the number of vertices in a batch. A boxplot of the target distribution can be seen in Figure 3. One can see that each distribution shows a substantial fraction of outliers. The Root Mean Squared Error (RMSE) and relative errors are presented for each geometrical property. The relative error is calculated as

$$\mathbb{E}_{rel} = \sqrt{\frac{\sum_i^n (\hat{y}_i - y_i)^2}{\sum_i^n y_i^2}}. \tag{3}$$

We achieve errors in the order of 1e-2 for most properties, as can be seen in Table 1.

We further assess our model's performance on two more complex tasks: Building footprint classification and cartographic generalisation.

3.2 Building Classification

After establishing our methodology for learning Geometric Features directly from vector data, we now demonstrate its practical application in the important geographical task of Building Classification, where automated recognition of building types can significantly enhance urban planning, asset management, and cartographic processes.

For classification, we stack five of the described local Feature Extraction blocks (Figure 2) on top of each other and perform Maxpooling. Batch normalization layers follow after each convolutional layer, except in the first Feature

Geometrical Properties	RMSE	Relative Error
LengthInf	0.0150	0.0015
LengthSup	0.0149	0.0017
LengthBase	0.0189	0.0015
Cosine+1	0.0049	0.0146
Sine+1	0.0037	0.0133
Area	1.6682	0.1482
Height	0.0131	0.0035

Table 1. Root Mean Squared Error (RMSE) and relative errors of the proposed CNN based Feature Extraction Block for predicting local geometrical properties of building footprints.

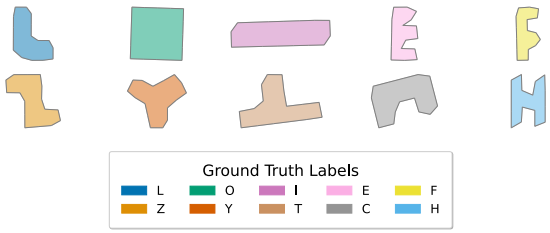

Figure 4. Randomly selected building footprints of each class. The colours indicate predicted class labels from our model, with ground truth values as shown in the legend.

Extraction block (Ioffe and Szegedy, 2015). ReLU activations are used throughout the network and output logits are produced by a fully connected layer. We train the model by minimizing its Cross Entropy, averaged over mini batches of size 32. The specifications are given in Table 2. We use the dataset from Yan et al. (2021) and follow the setup from Liu et al. (2021), to split the data into training and test set with a ratio of $8:2$. Liu et al. (2021) resample the building footprints to a fixed length of $L = 16$ vertices per footprint, using the Douglas-Peucker Algorithm (Douglas and Peucker, 1973). In practice, we also observe much better results with resampled data, compared to a minimum number of vertices per footprint and follow this protocol. We set the learning rate to $\lambda_0 = 0.001, \lambda_{min}$ =1e-5 and train for 150 epochs.

We report accuracy, precision, recall, and macro F1-score, averaged across $n = 5$ experimental runs for robust evaluation. Our model significantly outperforms previous approaches (Table 3). Figure 4 illustrates examples of correctly classified building footprints.

3.3 Cartographic Generalisation

Building on our Feature Extraction Block described in Section 3.1, we now turn to a much more challenging and practical application in digital cartography: Cartographic Generalisation. This process traditionally required intensive manual work and rule based algorithms. In order to solve this task we will require deeper models that can effectively account for the abundant variations seen in Build-

Layer Name	**Output Dimension**
Input Layer	$2 \times L$
Block1 (No norm)	$256 \times L$
Block2	$256 \times L$
Block3	$512 \times L$
Block4	$512 \times L$
Block5	$1024 \times L$
MaxPool	$1024 \times L$
Flatten	1024
FC1	*n_classes*

Table 2. Specification of the network architecture for Building classification. On the right the output dimension of the respective layer is shown, where $L = 16$ denotes sequence length of vertices. Batch dimensions are omitted for simplicity. FC1 is a linear layer that outputs class logits.

Proceedings of the International Cartographic Association, 7, 26, 2025 | https://doi.org/10.5194/ica-proc-7-26-2025
32nd International Cartographic Conference (ICC 2025), 17–22 August 2025, Vancouver, Canada.
This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

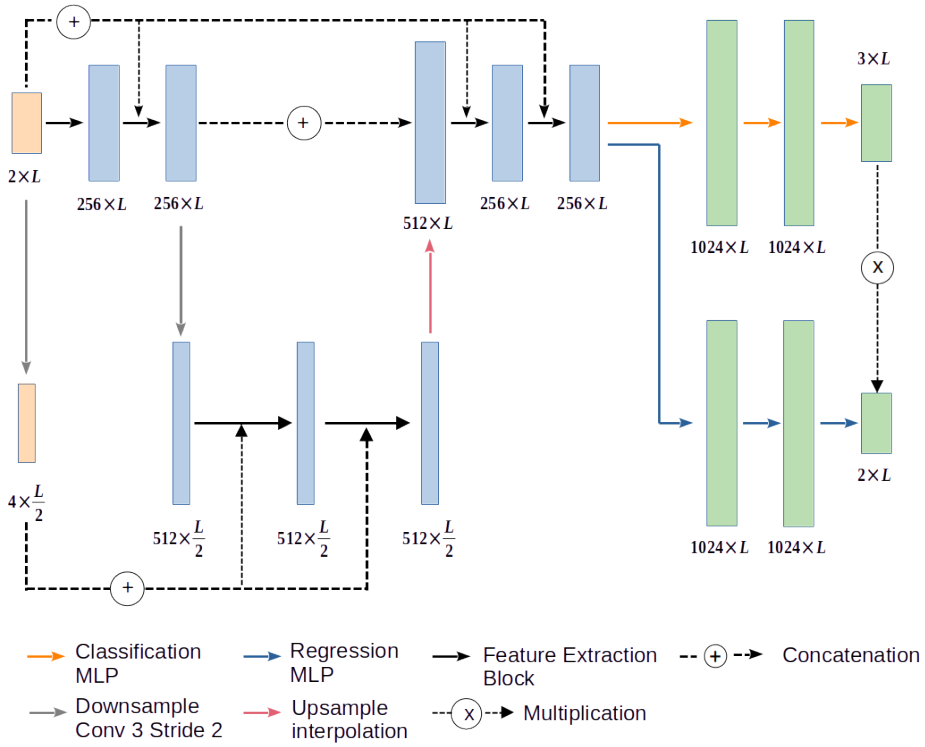

Figure 5. Architecture of Unet like model for Building Simplification. L denotes the vertex sequence length of a given footprint. Orange blocks (left) show the input coordinates and their linearly downsampled version. Blue blocks in the center show intermediate feature activations with the respective dimensions. Green blocks (right) show the task specific activation maps for regression and classification. A label dependent mask is multiplied to the regression outputs, setting non-move nodes to zero. Triangle Feature Extraction Blocks are implemented with residual connections, adding their inputs to the outputs. More information is given in the text.

ing Simplification. Furthermore we heavily rely on our self-supervised pretraining task from Section 3.1, enabling the model to extract Geometric Features.

We adopt the Building Simpification task from Zhou et al. (2023), as a combined node classification and regression problem. In this formulation, the simlplified building polygons are obtained from either clssifying a vertex as remove, keep or move. Displacement vector magnitudes within the node's local reference frame with respect to it's both neighboring vertices are regressed by a separate task head. To this end, we use the previously described building blocks (Figure 2) in an Encoder-Decoder structure, similar to Unet (Ronneberger et al., 2015). As this model is considerably deeper than the previous ones, we also use residual connections that add the input of each block to it's output, as well as adopting a bottleneck design, reducing the channel dimension before each convolution of size 3 (He et al., 2016). Furthermore, we add a Dropout layer with $p = 0.2$ for the MLP block (Srivastava et al., 2014).

Metrics	Model	
	Our Method	DPCN
Accuracy	**0.9919**	0.9842
Precision	0.9331	**0.9902**
Recall	**0.9980**	0.8376
F1 score	**0.9582**	0.8874

Table 3. Results for Building Classification. Mean values are reported over n=5 experiments and compared to results in DPCN (Liu et al., 2021).

In the encoder path, the spatial dimension is reduced by a fraction of $\frac{1}{2}$, after the second Feature Extraction Block. In the decoder path, the original spatial dimension need to be restored by upsampling. We use a linear convolutional kernel of size 3 and circular padding, where we sample every second node activation (e.g. with stride 2) for downsampling and linear interpolation for upsampling. Feature activations of the encoder path are concatenated to the decoder path. Furthermore, we concatenate input coordinates to each block, guiding the model to learn geometrically meaningful representations. In order to concatenate the coordinates to the downsampled feature activations, we also directly downsample the footprint coordinates in the same way. After the last block, the final feature activations are then processed by task specific heads, implemented as 2-layer MLPs with dropout after each activation with rate $p = 0.2$. Finally, as in (Zhou et al., 2023), regression outputs are set to zero which are not classified as move nodes by multiplication with a mask that depends on the predicted class labels. This creates an imbalance in the dataset, which we counteract, by weighting the loss functions proportionally to the class frequencies. We use Cross Entropy for classification and MAE for regression. Furthermore, we employ group normalization with a group

This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

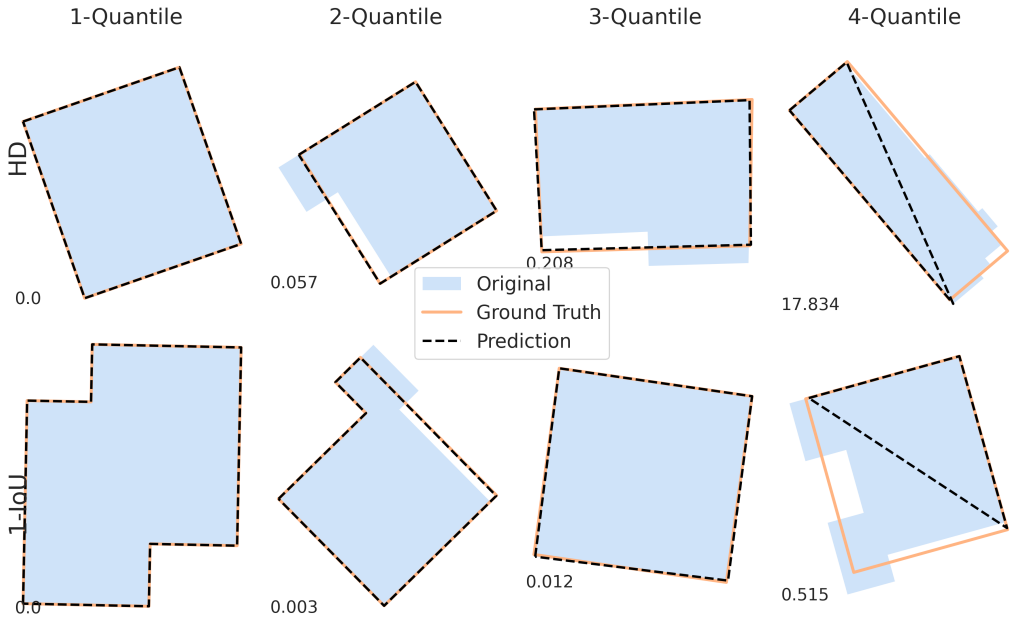

Figure 6. Quantiles for 1-IoU and Hausdorff Distance(HD) of reconstructed building footprints are shown. With corresponding values for each metric depicted on the bottom left corner.

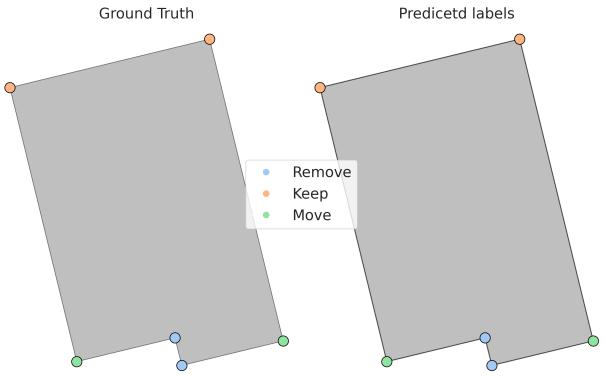

Figure 7. Example of Ground truth and predicted node labels on the Building Simplification task.

size of 16 and ReLU's throughout the network (Wu and He, 2018). In total our model comprises six Feature Extraction blocks, followed by two task-specific MLP's of 2 layers each. The architecture can be seen in Figure 5.

Such deeper models require large datasets to effectively tune the vast number of parameters. However, high quality annotated data is scarce. One promising approach is self-supervised learning, where the model is first trained on a different but related pretext task (Wang et al., 2022). The model's learned representations can then be leveraged for the downstream task at hand. In our case, we use the learned Geometric Embeddings from the regression task (Section 3.1) to pretrain the model in a self-supervised fashion before training on the Cartographic Generalisation task.

We use the dataset of simplified buildings in the City of Stuttgart from Zhou et al. (2023), which was first descibed in (Feng et al., 2019). The dataset consists of a total of 8494 building footprints, with corresponding ground truth simplifications. These are split into train-,test- and validation set with a ratio of 6:2:2, respectively. Despite the relatively small size of the dataset, we observe satisfactory results even for relatively deep models, due to our self-supervised approach. The learning rate is annealed in the range of [1e-4,1e-6] and we train for 500 epochs.

We report MAE for regression, as well as Accuracy and macro-averaged F1-score for node classification. Further-

Model	Remove		PreMove	NextMove
	Accuracy	F1-Score	MAE	MAE
CNN	0.871	0.836	0.242	0.241
CNN+features	0.868	0.836	0.241	0.240
CNN+pretraining	**0.904**	0.873	**0.236**	**0.235**
GCN	0.481	0.395	0.278	0.277
GraphSAGE	0.630	0.557	0.278	0.278
SplineCNN	0.708	0.614	0.2829	0.2682
Zhou et al.	0.862	**0.926**	0.390	0.388

Table 4. Results on the cartographic generalisation task, reporting MAE for regression and Accuracy and macro-averaged F1-Score for node classification. Results of our proposed CNN based model are compared to similar GNN archiectures (GCN,GraphSAGE, SplineCNN), as well as results from Zhou et al. (2023).

Proceedings of the International Cartographic Association, 7, 26, 2025 | https://doi.org/10.5194/ica-proc-7-26-2025
32nd International Cartographic Conference (ICC 2025), 17–22 August 2025, Vancouver, Canada.
This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

more, we compare our CNN based framework to several architectures based on Graph Convolutions. To this end, we replace the convolutional layer of kernel size 3 in the previously described Feature Extraction Block (Figure 2), with a Geometric Convolution and stack this block four times. Specifically, we test the Graph Convolutional Layer (GCN) (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017) and SplineCNN (Fey et al., 2018) and also use these models to directly train on vector coordinates. We present further ablation studies by enhancing the input feature vector with differences to neighboring vertices as in (Mai et al., 2023, Liu et al., 2021), as well as evaluating the performance of the proposed model without pretraining. Besides, we show the results from Zhou et al. (2023) for better comparison. The results are summarized in Table 4, highlighting the efficacy of purely CNN based deep models even on geometric tasks such as Cartographic Generalisation. Additionally, we calculate Intersection over Union (IoU) and Hausdorff distance (HD) between predicted and ground truth reconstructions. Quantiles of these metrics are shown in Figure 6.

4. Discussion

We investigated CNN-based modules for building polygons, demonstrating that our Feature Extraction blocks can learn Geometric Embeddings directly from vector coordinates (Table 1). This approach, when stacked in a Building Classification task, outperformed previous methods (Table 3). For Building Simplification, our Unet-like architecture with self-supervised pretraining surpassed GNN-based formulations on several metrics (Table 4).

Self-supervised pretraining proved valuable for deeper models requiring substantial data, while showing minimal benefit for shallower networks. Similarly, advantages of hand-designed features become negligible with increased network depth. This demonstrates the capacity of Deep Learning to extract geometric representations from raw coordinates without engineered features.

Our work reveals the effectiveness of CNN architectures for processing vector coordinates in constrained spaces, even outperforming comparable GNN-based approaches (Table 4). Unlike rasterization methods, our model operates directly on geometric data, preserving spatial relationships. This suggests that Convolutions effectively aggregate node features from neighboring vertices in building shape analysis, without the need of specialized architectures that incorporate geometrical knowledge.

For Building Simplification, our Unet-like model with effective self-supervised training surpassed the impressive results in Zhou et al. (2023) on several metrics (Table 4). Figure 6 shows nearly half the dataset with zero error and up to 75% with distances to Ground Truth reconsructions close to zero. Nevertheless, some outliers occur when nodes are mislabeled as "remove." Training schemes specifically targeting this issue could potentially improve results.

While our results suggest that our models learn Geometric Features effectively, CNNs remain dependent on shifts in irregular vector spaces. Results from Liu et al. (2021) indicate equidistant sampling to a constant number of vertices improves performance by generating a regularly sampled grid on the polygon's boundary. This appropriate inductive bias for CNNs could also benefit tasks like Building Simplification.

Despite our advances, some geometric properties remain challenging for our models (Table 1). Future work could explore self-supervised training for more advanced architectures like Transformers (Vaswani et al., 2017), potentially offering greater efficiency through larger batch training.

5. Conclusion

This work introduces a CNN-based model for processing vector geometries directly from 2D coordinates, offering an efficient and scalable alternative to existing methods. By avoiding hand-crafted features and complex architectures, our approach enables end-to-end learning for a wide range of GeoAI applications, including Building Classification and Cartographic Generalisation. We explicitly make use of learning Geometric Embeddings in a self-supervised manner, leveraging the full potential of deep models even on tasks with little annotated data.

Our framework addresses several key challenges in applying Deep Learning to vector geographic data. First, by learning geometric properties through self-supervision, we eliminate the need for extensive manual Feature Engineering that has traditionally been required when working with vector representations. Second, our relatively simple CNN based Feature Extraction blocks provide competitive performance while maintaining computational efficiency. The effectiveness of our approach was demonstrated across multiple tasks of increasing complexity. From the initial regression of local geometrical properties to Building Classification and finally to the intricate problem of Cartographic Generalisation, our model consistently delivered robust performance. Future developments may include investigation of self-supervision for more complex architectures as well as integration with other types of geometrical vector data.

References

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P., 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34(4), pp. 18–42.

Douglas, D. H. and Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization* 10(2), pp. 112–122.

Feng, Y., Thiemann, F. and Sester, M., 2019. Learning cartographic building generalization with deep convolutional neural networks. *ISPRS International Journal of Geo-Information* 8(6), pp. 258.

Fey, M., Lenssen, J. E., Weichert, F. and Müller, H., 2018. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In: *Proceedings of the IEEE*

This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.

conference on computer vision and pattern recognition, pp. 869–877.

Groh, F., Wieschollek, P. and Lensch, H. P., 2018. Flex-convolution: Million-scale point-cloud learning beyond grid-worlds. In: *Asian Conference on Computer Vision*, Springer, pp. 105–122.

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

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning*, pmlr, pp. 448–456.

Kingma, D. P., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Liu, C., Hu, Y., Li, Z., Xu, J., Han, Z. and Guo, J., 2021. Triangleconv: A deep point convolutional network for recognizing building shapes in map space. *ISPRS International Journal of Geo-Information* 10(10), pp. 687.

Loshchilov, I. and Hutter, F., 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Mai, G., Jiang, C., Sun, W., Zhu, R., Xuan, Y., Cai, L., Janowicz, K., Ermon, S. and Lao, N., 2023. Towards general-purpose representation learning of polygonal geometries. *GeoInformatica* 27(2), pp. 289–340.

Pope, P., Zhu, C., Abdelkader, A., Goldblum, M. and Goldstein, T., 2021. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, pp. 234–241.

Simonovsky, M. and Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1), pp. 1929–1958.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*.

Veer, R. v., Bloem, P. and Folmer, E., 2018. Deep learning for classification tasks on geospatial vector polygons. *arXiv preprint arXiv:1806.03857*.

Wang, S., Suo, S., Ma, W.-C., Pokrovsky, A. and Urtasun, R., 2018. Deep parametric continuous convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2589–2597.

Wang, Y., Albrecht, C. M., Braham, N. A. A., Mou, L. and Zhu, X. X., 2022. Self-supervised learning in remote sensing: A review. *IEEE Geoscience and Remote Sensing Magazine* 10(4), pp. 213–247.

Wu, W., Qi, Z. and Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 9621–9630.

Wu, Y. and He, K., 2018. Group normalization. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Yu, P. S., 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32(1), pp. 4–24.

Yan, X., Ai, T., Yang, M. and Tong, X., 2021. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science* 35(3), pp. 490–512.

Zhou, Z., Fu, C. and Weibel, R., 2023. Move and remove: Multi-task learning for building simplification in vector maps with a graph convolutional neural network. *ISPRS Journal of Photogrammetry and Remote Sensing* 202, pp. 205–218.

Proceedings of the International Cartographic Association, 7, 26, 2025 | https://doi.org/10.5194/ica-proc-7-26-2025
32nd International Cartographic Conference (ICC 2025), 17–22 August 2025, Vancouver, Canada.
This contribution underwent single-blind peer review based on submitted abstracts. © Author(s) 2025. CC BY 4.0 License.