# GEOVEX: GEOSPATIAL VECTORS WITH HEXAGONAL CONVOLUTIONAL AUTOENCODERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We introduce a new geospatial representation model called GeoVeX to learn global vectors for all geographical locations on Earth land cover (200+ million embeddings). GeoVeX is built on a novel model architecture named Hexagonal Convolutional Autoencoders (HCAE) combined with a Zero-Inflated Poisson (ZIP) reconstruction layer, applied to a grid of Uber's H3 hexagons, each one described by the histogram of OpenStreetMap (OSM) geographical tags occurrences. GeoVeX is novel on three aspects: 1) it produces the first geospatial vectors trained on worldwide open data, enabling wide adoption on every downstream tasks which may benefit from enriched geographical information, requiring only location coordinates; 2) it represents the first use of hexagonal convolutions within autoencoder architectures, to learn latent representations of an hexagonal grid; and 3) it introduces a spatial-contextual Poisson reconstruction loss function for autoencoder architectures suitable for training on sparse geographical count data. Experiments demonstrate that GeoVeX embeddings can improve upon state-of-the-art geospatial location representations on two different downstream tasks: price prediction in the travel industry and hyperlocal interpolation of climate data from weather stations.

## 1 INTRODUCTION

Entity embedding is ubiquitous in a variety of Machine Learning tasks thanks to its many advantages: it captures the semantics of each entity in the context of a given domain; it enables transfer learning to different related tasks; it reduces the sparsity of the entity representation and compresses the feature space. In NLP domain, global word embedding models, such as *Word2Vec* (Mikolov et al., 2013), *GloVe* (Pennington et al., 2014) and *BERT* (Devlin et al., 2019) have been successful at capturing the word semantics of big open-source vocabularies (e.g. Wikipedia, Gigaword) and are used to transfer learning to multiple downstream tasks, such as sentiment analysis (Tang et al., 2014; Deho et al., 2018; Alamoudi & Alghamdi, 2021), question retrieval (Zhou et al., 2015), and medical semantics (Wang et al., 2018). Similar approaches inspired by NLP have been since then proved to be useful in many industrial domains, where multiple models have been proposed for learning the latent representations of entities specific to an industry, such as *Product2Vec* (Biswas et al., 2017) and *User2Vec* (Hallac et al., 2019) in e-commerce, or *Wave2Vec* (Baevski et al., 2020) in speech representation, just to name a few.

In comparison, in the field of Geographic Information Science (GIS), a global set of task-agnostic embeddings for geographical space representation can benefit multiple domains and use cases, such as: price prediction for houses (Wang et al., 2021), hotel rooms (Kisilevich et al., 2013), and vacation homes (Islam et al., 2022; Pradip & Suthar, 2022); interpolation of climate variables such as temperature and pressure (Wu & Li, 2013); computer vision tasks with geo-located images (Berg et al., 2014). These tasks, just to name a few, have in common the application of some transformations to the spatial coordinates, but they do not leverage the spatial distribution of geo entities (such as parks, water, beach, buildings, streets, bars, etc.), which convey a more rich information of the geographical context. Besides, in terms of modelling, previous approaches to learn geospatial embeddings have a set of limitations, such as being non-contextual, task-specific and/or region-specific (see Sec. 2) that we address with a novel model architecture and loss function formulation (Sec. 3.6).

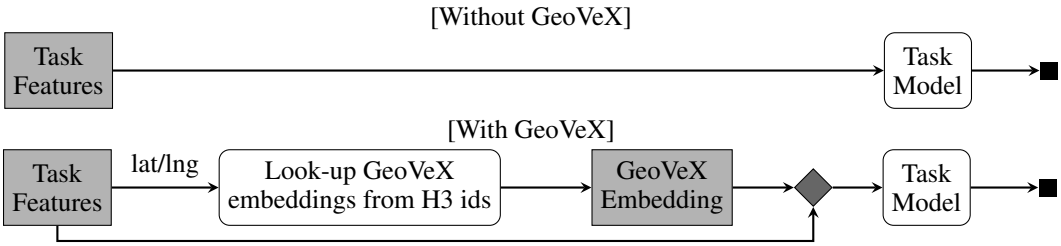


Figure 1: Workflow to use GeoVeX embeddings in downstream tasks: the feature set is expanded by concatenating initial task features with the GeoVeX embedding associated to the H3 hexagon corresponding to the latitude (lat) and longitude (lng) coordinates.

Our new approach to learn the geospatial embedding of each location on Earth, based on nearby geographical entities, aims to pre-train a finite set of embeddings that can cover the whole Earth, so that they can be stored and used with ease as extra features in any Machine Learning task where entities have latitude and longitude coordinates. The general workflow is summarized in Fig. 1. To achieve the goal of wide adoption to multiple downstream tasks, we leverage Uber Hexagonal Hierarchical Spatial Index grid system named H3[1] to spatially index the data coordinates into small regions of approximately the same size, since H3 minimizes map distortion. A pair of coordinates $(i, j)$ is thus represented by a unique H3 hexagonal id for which we learn a GeoVeX embedding.

To learn GeoVeX embeddings for all the H3 hexagons, we associate each one of them to the geographical tags of the entities obtained from OpenStreetMap (OSM)[2]. OSM is a project that creates and distributes free worldwide geographic data, and, as of January 2022, has ≈7 billion nodes and ≈4 million map changes per day. This makes it the equivalent of Wikipedia for word embeddings: a massive, scalable, and information-rich global open dataset for creating and updating global embeddings. In particular, OSM contains nodes, ways and relations, which together can be transformed to points, lines and polygons, each one characterized by a set of semantic tags, such as *amenity:bar*, *highway:motorway*, *natural:forest*. By intersecting these OSM entities with the H3 hexagons, and by using a Bag Of Words (BOW) model on their tags, each hexagon can be sparsely described by a $K$-dimensional histogram vector, where $K$ is the size of a subset of the vocabulary of OSM geographical tags, where each element represents the number of times an entity with the respective tag is contained or intersects the hexagon itself. This information needs then to be properly aggregated to produce an embedding while, at the same time, taking into account the information from neighboring hexagons, which provide the geographical context. This concept follows the first law of geography: *everything is related to everything else, but near things are more related than distant things* (Tobler, 1970). The convolution operation, borrowed from Computer Vision, presents in this domain challenges to address: 1) not square, but hexagonal grids, and 2) different distribution of each "channel": not dense pixel values, but highly sparse counts. GeoVeX model aims to bridge the gap of convolutional neural networks usage on these hexagonal grids described by sparse counts.

In summary, the **contributions** of our work are threefold:

1. the GeoVeX model design, to learn task-agnostic global geospatial vectors from worldwide open GIS databases. The produced embedding vectors can effectively be leveraged to improve predictions in downstream tasks which involve geographical entities designated by their coordinates. We demonstrate their expressive power quantitatively for two downstream tasks on public datasets in different domains and parts of the world: price prediction in the US travel industry (Sec. 4.1) and temperature interpolation of climate data from Italian weather stations (Sec. 4.2);

2. the first Hexagonal Convolutional Autoencoder (HCAE) network, to demonstrate that standard and transposed hexagonal convolutions can be successfully coupled with an autoencoder network architecture to learn latent representations of hexagonal grids (Sec. 3.4);

[1]https://eng.uber.com/h3

[2]https://www.openstreetmap.org/

3. we introduce the first Zero-Inflated Poisson (ZIP) autencoder's probabilistic decoder block, which is trained with a spatial contextual Poisson loss function, to adapt the standard reconstruction layer of autoencoders network to the case of zero-inflated spatial contextual count data produced by the H3 grid and the OSM entity tags counts (Sec. 3.6).

## 2 Related Work

### 2.1 Global geospatial embeddings

*Space2Vec* (Mai et al., 2020) and *Hex2Vec* (Woźniak & Szymański, 2021) are the closest approaches to our work.

Space2Vec (Mai et al., 2020) is one of the state-of-the-art methods proposed to encode a pairs of coordinates in a high dimensional embedding which can be used in downstream tasks. In particular it uses trigonometric functions with different frequencies to model a given position in space through dimensional expansion. The approach can actually be considered as a worldwide embedding model in the sense that any coordinate on Earth can be transposed to a latent representation. However, the embeddings are not learnt based on the nearby geographical entities (e.g. buildings, streets, bars, etc.), so Space2Vec vectors cannot be really considered as pre-trained geospatial embeddings aiming at transfer learning, in which the vectors are used to draw similarities of different regions on Earth based on geographical characteristics.

Hex2Vec (Woźniak & Szymański, 2021) is a state-of-the-art work that uses H3 hexagons described by OSM entities, and develops an approach to learn geospatial embeddings. GeoVeX follows a similar approach to create the training setup (H3 as grid methodology, and OSM as provider of geographical entities), but it is different in terms of both model design and loss function formulation. In fact, Hex2Vec proposed a skip-gram model with negative sampling, paired with a triplet loss function: for each hexagon to embed (the triplet *anchor*), the *positive* hexagon is taken from the first adjacent ring of hexagons, and the *negative* hexagon is sampled from at least 3 rings away from it. Even if the first law of geography (Tobler, 1970) is used to justify the skip-gram design, it is easy to see that imposition of complete similarity between the hexagon to embed (the anchor) and its adjacent ones (the positives) in the loss formulation is not always appropriate since it imposes perfect similarity of a beach-like hexagon's embedding with a town-like one in case they are adjacent.

### 2.2 Local or task-specific geospatial embeddings

Other approaches to learn geospatial embeddings in GIS domain are trained for specific tasks and/or specific regions, so they cannot be used as pre-trained global geospatial embeddings: *Region2Vec* (Xiang, 2020) combines Point Of Interest (POI) and mobile sensors data obtained in a specific region in China; *Zone2Vec* (Du et al., 2018) uses trajectories generated by taxis in Beijing; *Urban2Vec* (Wang et al., 2020) uses Street View images and focuses only on the urban use cases, where such images are available; *RegionEncoder* (Jenkins et al., 2019) combines mobility from taxis, categories of POI and satellite images to create a model specific to two urban regions; *Tile2Vec* (Jean et al., 2019) applies an unsupervised representation learning on satellite images, while using the triplet loss formulation, which leads to the same issues described for Hex2Vec; in (Islam et al., 2022), the authors apply Moran Eigenvector Spatial Filtering (MESF) for the specific task of predicting AirBnB listing price in San Jose County in US.

### 2.3 Convolutional Autoencoders

In terms of network architecture, our design is inspired by recent works on Convolutional Autoencoders (CAE), among which the deep CAE architecture of Guo et al. (2017) represents an example of the state of the art. The authors propose the convolutional autoencoders to learn a latent representation in the bottleneck layer. However, the architecture is designed to work with images, and not with tensors that represent hexagonal shaped data nor to fit zero-inflated count variables, so we propose some modifications to account for this. In terms of loss function, the authors use a combination of reconstruction loss and clustering loss, but this would require apriori knowledge of the number of clusters in the data, which is feasible in their case (10 handwritten digits), but it is not straightforward in case of worldwide geospatial embeddings.

# 3 PROPOSED APPROACH: GEOVEX

In this section, first, we formalize the problem of training geospatial embeddings of a pair of coordinates on Earth, then we present the proposed GeoVeX model architecture, its input format and its novel specific components: hexagonal convolutions, the zero-inflated Poisson reconstruction layer for the autoencoder, and the loss function used to train the network.

## 3.1 PROBLEM FORMALIZATION

Given a pair of latitude and longitude coordinates on Earth $p = (x, y) \in \mathbb{R}^2$, define a function $F_\theta(p) : \mathbb{R}^2 \to \mathbb{R}^D$ parameterized by $\theta$ which maps the point $p$ to a $D$-dimensional embedding $e = F_\theta(p)$, which represents the geographical characteristics of the space nearby the given point proportionally to the distance to it, in accordance to Tobler (1970).

In this work, we propose to formulate the $F$ function as $F_\theta(p) = f_\theta(t(p), \mathcal{N}_{n'}(t(p)))$, where $f_\theta : \mathbb{R}^K \times \mathbb{R}^{n' \times K} \to \mathbb{R}^D$ is the GeoVeX encoder, and $t : \mathbb{R}^2 \to \mathbb{R}^K$ is the function which assigns the vector of counts $z = t(p)$ of $K$ geographical tags of the entities that intersect the hexagon where the point $p$ resides, and $\mathcal{N}_{n'}(t(p))$ is the set of count vectors of $n'$ neighbor hexagons of $p$. The GeoVeX pre-trained embedding $e$ can later be concatenated to existing features of any downstream task involving entities described by latitude and longitude coordinates. This effectively serves to expand the features set used by the model addressing the machine learning task to solve.

## 3.2 GEOVEX MODEL 3D TENSOR INPUT FROM 2D HEXAGONAL CELLS

In this section we briefly describe the GeoVeX 3D input tensor required by the network architecture. In the image domain, the first 2 dimensions represent the 2D coordinates and the 3rd dimension represents the color channels, and for each combination of coordinates and channel the color intensity value is defined. By analogy, in GeoVeX, the first 2 dimensions represent the 2D coordinates of the hexagon in a spatial coordinate system and the 3rd dimension describes the count of $K$ OSM tags of the entities intersecting a given H3 cell, following the Bag Of Words (BOW) model. After cleaning and preprocessing, we cut the number of tags to the top $K = 1024$ most frequent ones, since the vocabulary presents a long tail distribution. More details can be found in Section A.4 of Appendix.

Each hexagon's neighborhood is defined by the $r$-rings neighboring cells as described in Fig. 2 for $r = 7$ rings around the point $(0, 0)$ ($r = 7$ has been experimentally defined to fit our GPU memory constraints). The importance of each ring should decay with distance to center, in alignment with the first law of geography (Tobler, 1970). Hence, we propose an intuitive distance weighting kernel in the model loss function formulation in Sec. 3.6, so that rings that are far away have less impact on the learnt representation.

Transposition of the 2D hexagonal grid to 2D matricial form, for the first 2 dimensions of the 3D-tensor, is done via *axial* coordinate system representation (see Luo et al. (2019) for more details on different transposition methods), and is depicted in Fig. 3. The final 3D representation is $(2r + 1) \times (2r+1) \times K$. To enable correct convolutions within this network design, the dimensions of the input matrix are required to be a power of 2, which can be achieved through padding. Since we trained the network with $r = 7$ the input of the GeoVeX model is a $16 \times 16 \times 1024$ 3D tensor.

## 3.3 MODEL ARCHITECTURE

The architecture of GeoVeX network is summarized in Fig. 4. The base design is the one of a Convolutional AutoEncoders (CAE) network Guo et al. (2017), which is composed of two main components: the encoder $h = f(z)$ and the decoder $z' = g(h)$, where $z$ is the input tensor, $h$ is its latent representation learnt by applying the encoder block, and $z'$ is the reconstructed tensor created by the decoder block. The parameters of the CAE network are updated by minimizing the reconstruction error described in Sec. 3.6.

The encoder is a set of convolutional blocks stacked on the 3D input tensor to extract hierarchical features. The output units of the convolutions are flattened to form a vector, and are passed to a fully connected layer with $D$ units, the final dimension of GeoVeX embedding. In practice, we have found that $D = 32$ is a good trade-off between expressiveness and storage cost of the vector. The

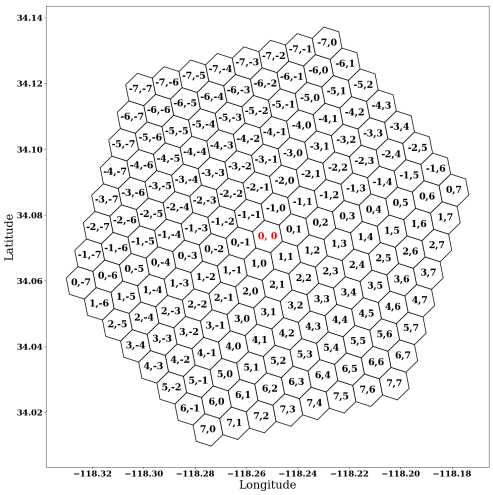


Figure 2: The hexagonal grid of the $r$-rings centered on the hexagon to embed, with the axial coordinates $(i, j)$ assigned to each hexagon based on its position in the grid.

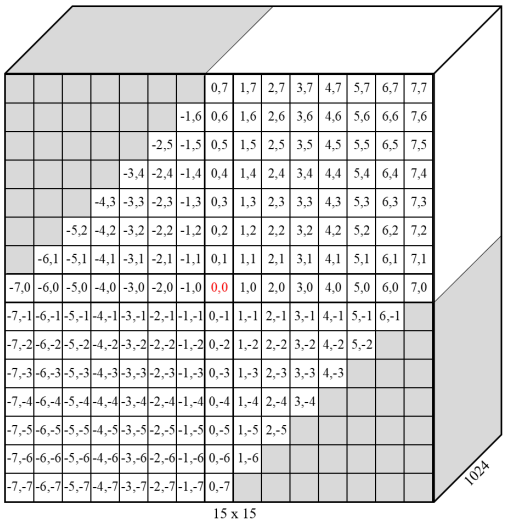


Figure 3: The transposed hexagonal grid represented as a 3D matrix, using the axial coordinates transposition and assigning the OSM counts as 1024 tensor channels. In grey, the padding cells that are not valid.

input tensor is thus transformed by the encoder into a 32 units latent embedding representation $h$, forcing the autoencoder to capture the most salient features that minimize the loss function. The decoder is defined as a set of convolutional transposed layers to transform embedded feature back to original input, followed by a novel Zero-Inflated Poisson layer of our contribution to reconstruct the count tensor in input (see Sec. 3.5 for motivation). Batch Normalization Ioffe & Szegedy (2015) is used to re-center and re-scale the counts tensor in input and output of every convolution operation. The usage of Batch Normalization and ReLU is aligned to Ioffe & Szegedy (2015): we apply the normalization before the non-linearity and we remove the redundant bias term in the convolution operation. In terms of up/sub sampling, GeoVeX uses convolutional layers with stride of 2, instead of convolutional layers followed by pooling layers, in both the encoder (standard convolution) and the decoder (transposed convolution, sometimes referred as "deconvolution"). In this way, the network can learn spatial subsampling and upsampling from data, leading to higher capability of transformation (Guo et al., 2017).

The GeoVeX network design presents two important characteristics:

1. The network represents the first Hexagonal Convolutional Autoencoders (HCAE) network, where the traditional CAE network design is adapted to work with hexagonal data representation and convolutions;

2. The network combines a Zero-Inflated Poisson (ZIP) probabilistic reconstruction layer with a respective Poisson loss function, which enables the learning of the HCAE network in the context of zero-inflated input data.

### 3.4 Hexagonal Convolutions

The main building block of both the encoder and the decoder is the hexagonal convolution operator, which aims at extracting relevant geographical features at different granularity. The network assumes in input an hexagonal grid transposed to a two dimensional matrix using the the *axial* coordinate system representation as described in Sec. 3.2. In this representation, unused matrix cells are put in the corners, avoiding empty cells in the middle of the matrix and supporting the hexagonal convolutional autoencoder network described below. Following this representation, the hexagonal convolution is achieved by multiplying the 3x3 convolution kernel with a 3x3 mask to set to zero

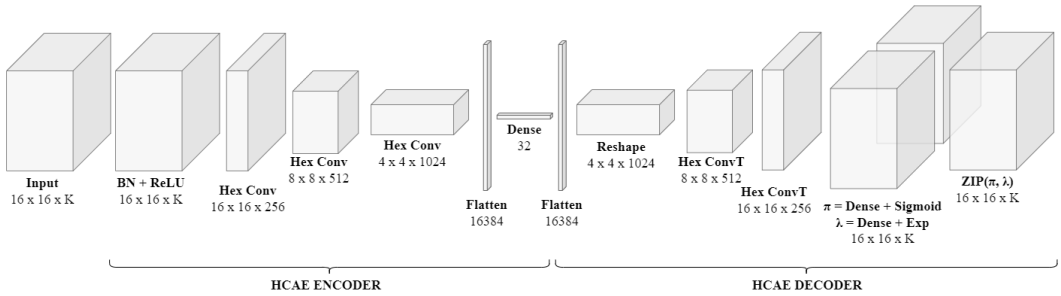


Figure 4: Illustration of the GeoVeX network architecture, showing the different layers of the Hexagonal Convolutional Autoencoders (HCAE) neural network with a Zero-Inflated Poisson (ZIP) probabilistic output layer to reconstruct the hexagonal matrix with geographical entity tags count. Each hexagonal convolution block and hexagonal transposed convolution block is composed by the respective convolution operator followed by Batch Normalization and ReLU operations.

the kernel at top-left and bottom-right cells (the mathematical formula of the 3x3 masked hexagonal convolution is given in Section A.2 of Appendix).

### 3.5 Zero-Inflated Poisson Reconstruction Layer

Since the network is intended to be used with geographical count data assigned to each hexagon, it expects a large number of zeros, since it is evident that for any given hexagon on Earth, only few geographical tags from the full vocabulary of tags are present. Multiple studies have demonstrated that the best fit to such data with excess of zeros is provided by zero-inflated distribution (Ridout et al., 1998; Beckett et al., 2014; Lambert, 1992; Unhapipat et al., 2018).

Among the set of zero-inflated distributions, we propose the use of the Zero-Inflated Poisson (ZIP), which is denoted by $ZIP(\pi, \lambda)$, has two parameters $\pi$ and $\lambda$, and has the following probability mass function, where $0 \leq \pi \leq 1$ and $\lambda \geq 0$:

$$\forall i \in \{1, \dots, K\}, \mathbb{P}(z_{.,.,i} = c) = \begin{cases} \pi + (1-\pi) \cdot e^{-\lambda} & \text{if } c = 0 \\ (1-\pi) \cdot e^{-\lambda} \cdot \frac{\lambda^c}{c!} & \text{if } c \in \{1, 2, 3, \dots\} \end{cases} \tag{1}$$

In the GeoVeX model, the parameters $\pi$ and $\lambda$ are estimated in the last layer of the decoder, and are used by the above ZIP mixture model to reconstruct, from the H3 embedding, the OSM count of each geographical tag $k$ of each hexagon at axial coordinates $i$, $j$. In fact, the final decoder layer produces two output variables for each coordinate $i, j, k$: $g_\pi$ and $g_\lambda$. The first one is activated with a *sigmoid* function to produce $\pi$ (Lambert, 1992), and the second one is activated with the *exponential* function to ensure that $\lambda > 0$, as is obviously required:

$$\widetilde{\pi}_{i,j,k} = \frac{e^{g_{\pi|i,j,k}(h)}}{1 + e^{g_{\pi|i,j,k}(h)}} \tag{2}$$

$$\widetilde{\lambda}_{i,j,k} = e^{g_{\lambda|i,j,k}(h)} \tag{3}$$

where $g_{\pi|i,j,k}(h)$, $g_{\lambda|i,j,k}(h)$ are the two output layers of the decoder network which are conditioned on embedding $h$ and produce the estimated $\widetilde{\pi}$ and $\widetilde{\lambda}$ for each coordinate $i, j, k$ (see Fig. 4).

### 3.6 GeoVeX Loss Function

GeoVeX proposes a novel adaptation of a Negative Log-Likelihood (NLL) loss function in the context of a hexagonal grid, to keep accordance with the first law of geography (Tobler, 1970). The proposed loss formulation encourages first the learning (and, thus, the reconstruction) of the geographical tags count describing the hexagon to embed, and then decay the importance of learning (and, thus, reconstructing) the tags count in each external ring of hexagons.

In details, the formulation of the GeoVeX loss function is an adaptation of the NLL loss with Zero-Inflated Poisson distribution, in the context of a spatial relationship among the neighbouring regions.

Based on the ZIP mass function (Eq. 1), and given that we have $N$ samples in each learning batch, the log-likelihood function of each element of the 3D reconstructed tensor can be derived as:

$$\begin{aligned}\log \mathcal{L}(\widetilde{\pi}, \widetilde{\lambda}; y)_{i,j,k} = &\sum_{n=1}^{N} \mathbb{I}(y_{n|i,j,k} = 0) \ln[\widetilde{\pi}_{n|i,j,k} + (1 - \widetilde{\pi}_{n|i,j,k}) e^{\widetilde{\lambda}_{n|i,j,k}}] \\ &+ \sum_{n=1}^{N} \mathbb{I}(y_{n|i,j,k} > 0)[\ln(1 - \widetilde{\pi}_{n|i,j,k}) - \widetilde{\lambda}_{n|i,j,k} + y_{n|i,j,k} \ln \widetilde{\lambda}_{n|i,j,k} - \ln(y_{n|i,j,k}!)]\end{aligned} \tag{4}$$

where $n$ is the n-th sample, $i$ and $j$ are the two axial coordinates, $k$ is the $k$-th geographical tag count, and $\mathbb{I}(y_{n|i,j,k})$ denotes an indicator variable that takes value 1 if the condition is verified, and zero otherwise. More details on the derivation can be found in Hossain & Howlader (2015) and Lambert (1992). To make the loss of each grid hexagon weighting proportionally to the ring where it is located, we apply two positional weights to the ZIP NLL loss of each hexagon of the grid, based on its $i, j$ axial coordinates:

- the *distance weighting kernel* $W_{dist}$ decays the importance of each subsequent neighbouring $r$-ith ring of hexagons, based on the inverse of the distance from the central hexagon to the ring number $r$ (i.e. ring $r = 1$ applies a weight of 1/2, while ring $r = 7$ applies a weight of 1/8);
- the *numerosity weighting kernel* $W_{num}$ equalizes the importance of each hexagonal $r$-ith ring, by subdividing the loss among the hexagons that compose it, so that each ring has same total importance in the final loss. By construction, each ring is composed of 6 times the number of the ring number $r$ (i.e. ring $r = 1$ has 6 hexagons, while ring $r = 7$ has 42 hexagons, so each hexagon has to weight $6 \cdot r$ to make each ring equal in importance)

The mathematical formulations in Eq. 5 and 6 are defined by construction, based on the semantic imposed by the definitions above, where *r* is the hexagonal ring number, and $|i - j| \leq R$ is a simple condition to explicitly put to zero the loss at the invalid top-left and bottom-right values in the axial coordinates matrix representation.

$$w_{dist_{i,j}} = \begin{cases} 1 & \text{if } i = j = 0 \\ 1/(1 + r_{i,j}) & \text{if } |i - j| \leq R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$w_{num_{i,j}} = \begin{cases} 1 & \text{if } i = j = 0 \\ 1/(6 \cdot r_{i,j}) & \text{if } |i - j| \leq R \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The two loss weighting kernels are depicted in Fig. 7 and 8 in Appendix to help visualization of the kernel effect. By combining the two weighting kernels, the loss to minimize is the sum of the ZIP negative log-likelihood loss of each $k$-th geographical tag count of each $i, j$ element of the hexagonal grid (Eq. 4), weighted based on its grid position using the kernels $W_{dist}$ and $W_{num}$:

$$\begin{aligned} L &= -\sum_{k=1}^{K} \log \mathcal{L}(\widetilde{\pi}, \widetilde{\lambda})_k \cdot W_{dist} \cdot W_{num} \\ &= -\sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{M} \log \mathcal{L}(\widetilde{\pi}, \widetilde{\lambda})_{i,j,k} \cdot \frac{w_{dist_{i,j}}}{\sum_{i,j=1}^{M} w_{dist_{i,j}}} \cdot \frac{w_{num_{i,j}}}{\sum_{i,j=1}^{M} w_{num_{i,j}}} \end{aligned} \tag{7}$$

## 4 EXPERIMENTS

We conducted experiments on two downstream tasks to validate the GeoVeX global vectors: 1) prediction of a vacation rental listing price in the travel industry and 2) hyperlocal interpolation of

temperatures from weather stations. In both cases, GeoVex embeddings are pre-trained, agnostic of the downstream tasks, frozen and then used as-is as additional features. To evaluate the expressiveness of GeoVeX vectors, we perform the following ablation study by comparing models using:

1. Common task features (X - Baseline);
2. Additional features: GeoVeX vectors (X - GeoVex);
3. Additional features: Hex2Vec vectors (X - Hex2Vec);
4. Additional features: Space2Vec encoding vectors (X - Space2Vec);
5. Additional features: both GeoVeX and Space2Vec (X - GeoVeX & Space2Vec);
6. Additional features: both Hex2Vec and Space2Vec (X - Hex2Vec & Space2Vec);

Note that for fair comparison, Hex2Vec vectors have been trained on the same OSM data as for GeoVeX, and Space2Vec hyper-parameters have been optimized for best task performance.

**Semantic analysis** The analysis of the semantic captured by GeoVeX compared to Hex2Vec, when trained on same input data, for the sake of brevity, can be found in Section A.1 of Appendix, where we show that GeoVeX improves upon Hex2Vec.

### 4.1 Vacation Rentals Price Prediction Task

**Dataset and Task** Open-source Inside Airbnb dataset[3] contains 210,811 listings located in the United States and describes each vacation rental listing with home characteristics, geographical coordinates, name of the listing location neighborhood and daily price which is the value to predict. The hypothesis is that the geographical characteristics of the location influence its price.

**Compared models** All competing models are trained with the same Gradient Boosting Machines regression model architecture (GBM), which enable the discovery of interaction among features. The model hyperparameters are optimized independently for each model setup through random search. Relevance of using GBM architecture is justified by the poor performance of a simple Ordinary Least Squares (OLS) baseline linear model for which we also report results. In terms of features, we consider a baseline model trained on the listing features mentioned above (please see also (Islam et al., 2022)), where geographical information is only latitude, longitude and the name of the neighborhood in which the listing is located. Its intent is not be used as a performance benchmark, but to underline how much information the additional geospatial embeddings bring.

The loss function is the Mean Squared Error (MSE) of the logarithm of the price, and the metrics monitored are both the loss and the Mean Absolute Error (MAE) of the price. The listings are split into training, validation and test with ratios 80%;10%;10%.

**Results** Table 1 shows the performance of all models on two different test sets of unseen 1) coordinates and 2) neighborhoods. We can see that the GeoVeX vectors are useful in both test strategies. For unseen coordinates, the GBM model with GeoVeX vectors outperforms both Hex2Vec vectors and Space2Vec coordinates encoder. For unseen neighborhoods, the GBM model with GeoVeX vectors and Space2Vec encoded coordinates outperforms the two models where these are used alone, and the GBM model with GeoVeX vectors alone is the second best model. This demonstrates that GeoVeX model improves upon Hex2Vec to compress H3 and OSM information, and that GeoVeX vectors add geospatial information that is not possible to obtain by encoding the location coordinates without considering the geographical entities, as is the case for Space2Vec.

### 4.2 Hyperlocal Temperatures Interpolation Task

**Dataset and Task** ARPA Lombardia, a regional agency for environment protection provides open weather sensors data[4] in the area of Milan (Italy). The data contains 1.4 millions of hourly temperature samples, recorded by 13 weather stations from January 2010 to August 2022 in Milan and

[3] http://insideairbnb.com

[4] https://www.arpalombardia.it/Pages/Meteorologia/Richiesta-dati-misurati.aspx

Table 1: Experiment results for the price prediction task. The lower the better.

| Model | Coordinates | | Neighborhoods | |
|---|---|---|---|---|
| | MSE ln(Price) | MAE Price | MSE ln(Price) | MAE Price |
| Linear - Baseline | 0.254 | 75.67 | 0.377 | 84.46 |
| GBM - Baseline | 0.198 | 65.42 | 0.262 | 70.54 |
| GBM - GeoVeX | **0.190** | **63.74** | 0.242 | 68.82 |
| GBM - Hex2Vec | 0.195 | 64.70 | 0.252 | 69.08 |
| GBM - Space2Vec | 0.201 | 65.98 | 0.247 | 68.41 |
| GBM - GeoVeX & Space2Vec | 0.192 | 64.28 | **0.239** | **67.91** |
| GBM - Hex2Vec & Space2Vec | 0.195 | 64.77 | 0.249 | 68.80 |

nearby. Each sample is described by: the temperature, the date, the hour, and the coordinates of the weather station. The task is the interpolation of the temperature between weather stations, which is modeled as a prediction of the temperature in an unknown location. This is achieved by removing a subset of stations from the training data and by comparing the model predictions with the actual temperatures observed by the test stations. The hypothesis is that the geographical characteristics of the location help to explain the differences in recorded temperatures from nearby sensors.

**Compared models** The experimental setup is the same as in Sec. 4.1. Baseline model denotes the model relying on all the available features when geospatial data are only latitude and longitude which are the date components (year, month and week of year) and the hour of the day. The loss function is the MSE of the temperature, and the metrics monitored are both the loss and the MAE.

**Results** Table 2 shows the performance of the different models on the test set. We can see that the GBM model with GeoVeX vectors improves upon the baseline, and that, when combined to Space2Vec, the resulting model performs better than the one with Space2Vec alone. This shows that GeoVeX vectors help in improving the estimation of temperatures in this region with a mix of urban and green areas, by providing the geographical context of each location.

Table 2: Experiment results for the hyperlocal temperature interpolation task. The lower the better.

| Model | MSE Temperature | MAE Temperature |
|---|---|---|
| Linear - Baseline | 18.561 | 3.47 |
| GBM - Baseline | 11.144 | 2.65 |
| GBM - GeoVeX | 10.657 | 2.62 |
| GBM - Hex2Vec | 10.862 | 2.63 |
| GBM - Space2Vec | 10.880 | 2.64 |
| GBM - GeoVeX & Space2Vec | **10.651** | **2.61** |
| GBM - Hex2Vec & Space2Vec | 10.870 | 2.63 |

## 5 Conclusion

We presented GeoVeX, a representation model which learns geospatial vectors as latent representations of the location's geographical characteristics. GeoVeX is based on a novel architecture named Hexagonal Convolutional Autoencoder (HCAE) paired with a Zero-Inflated Poisson (ZIP) reconstruction layer to learn from hexagonal grids described with zero-inflated counts of the geographical tags of OSM entities that intersect them. Two experiments on real open data support the fact that the pre-trained GeoVeX embeddings can improve models in multiple domains, either alone or combined with coordinates encoder techniques learnt specifically for the task at hand. In future, we aim to analyze the impact of changing the H3 grid resolution, test different zero-inflated probabilistic distributions, explore the semantic captured by the learnt representation, compare generic vectors against task-specialized vectors, and tackle a broader set of tasks that involve geolocated entities, such as: climate-related tasks (e.g. effect of urbanization on temperatures), classifications of geolocated images, or ranking of geolocated items.

## REFERENCES


Eman Saeed Alamoudi and Norah Saleh Alghamdi. Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. *Journal of Decision Systems*, 30(2-3):259–281, 2021.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rylwJxrYDS`.

Sadie Beckett, Joshua Jee, Thapelo Ncube, Sophia Pompilus, Quintel Washington, Anshuman Singh, and Nabendu Pal. Zero-inflated poisson (zip) distribution: Parameter estimation and applications to model data from natural calamities. *Involve, a Journal of Mathematics*, 7(6):751–767, 2014.

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2019–2026, 2014.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012. ISSN ISSN 1533-7928. URL `http://www.jmlr.org/papers/v13/bergstra12a.html`.

Arijit Biswas, Mukul Bhutani, and Subhajit Sanyal. Mrnet-product2vec: A multi-task recurrent neural network for product embeddings. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 153–165. Springer, 2017.

B. Oscar Deho, A. William Agangiba, L. Felix Aryeh, and A. Jeffery Ansah. Sentiment analysis with word embedding. *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, pp. 1–4, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Jiahong Du, Yujun Chen, Yue Wang, and Juhua Pu. Zone2vec: Distributed representation learning of urban zones. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 880–885. IEEE, 2018.

Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*, pp. 373–382. Springer, 2017.

Ibrahim Riza Hallac, Semiha Makinist, Betul Ay, and Galip Aydin. user2vec: Social media user representation based on distributed document embeddings. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–5, 2019.

Shakhawat Hossain and Hatem Howlader. Estimation techniques for regression model with zero-inflated poisson data. *International Journal of Statistics and Probability*, 4:64, 2015.

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

Md Didarul Islam, Bin Li, Kazi Saiful Islam, Rakibul Ahasan, Md Rimu Mia, and Md Emdadul Haque. Airbnb rental price modeling based on latent dirichlet allocation and mesf-xgboost composite model. *Machine Learning with Applications*, 7:100208, 2022.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3967–3974, 2019.

Porter Jenkins, Ahmad Farag, Suhang Wang, and Zhenhui Li. Unsupervised representation learning of spatial data via multimodal embedding. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1993–2002, 2019.

Slava Kisilevich, Daniel A. Keim, and Lior Rokach. A gis-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers' context. *Decis. Support Syst.*, 54:1119–1133, 2013.

Diane Lambert. Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.

Junren Luo, Wanpeng Zhang, Jiongming Su, and Fengtao Xiang. Hexagonal convolutional neural networks for hexagonal grids. *IEEE Access*, 7:142738–142749, 2019.

Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and N. Lao. Multi-scale representation learning for spatial feature distributions using grid cells. *ArXiv*, abs/2003.00824, 2020.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Prajapati Pradip and Prof. Monali Suthar. A survey on price prediction model for airbnb listing using machine learning. *International Journal of Scientific Research in Science, Engineering and Technology*, 2022.

Martin Ridout, Clarice GB Demétrio, and John Hinde. Models for count data with many zeros. In *Proceedings of the XIXth international biometric conference*, volume 19, pp. 179–192. International Biometric Society Invited Papers Cape Town, South Africa, 1998.

Duyu Tang, Furu Wei, Nan Yang, M. Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, 2014.

Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

Suntaree Unhapipat, Montip Tiensuwan, and Nabendu Pal. Bayesian predictive inference for zero-inflated poisson (zip) distribution with applications. *American Journal of Mathematical and Management Sciences*, 37(1):66–79, 2018.

Pei-Ying Wang, Chiao-Ting Chen, Jain-Wun Su, Tingwei Wang, and Szu-Hao Huang. Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism. *IEEE Access*, 9:55244–55259, 2021.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20, 2018.

Zhecheng Wang, Haoyuan Li, and Ram Rajagopal. Urban2vec: Incorporating street view imagery and pois for multi-modal urban neighborhood embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1013–1020, 2020.

Szymon Woźniak and Piotr Szymański. hex2vec: Context-aware embedding h3 hexagons with openstreetmap tags. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pp. 61–71, 2021.

Tingting Wu and Yingru Li. Spatial interpolation of temperature in the united states using residual kriging. *Applied Geography*, 44:112–120, 2013.

Mingjun Xiang. Region2vec: An approach for urban land use detection by fusing multiple features. In *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, pp. 13–18, 2020.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL*, 2015.