International Journal of Geo-Information

Article

A Hierarchy-Aware Geocoding Model Based on Cross-Attention within the Seq2Seq Framework

Linlin Liang [1,2,3], Yuanfei Chang [1,3,*], Yizhuo Quan [1,2,3] and Chengbo Wang [1,3]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; lianglinlin21@mails.ucas.ac.cn (L.L.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] National Engineering Research Center for Geomatics, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China
* Correspondence: changyf@aircas.ac.cn

Abstract: Geocoding converts unstructured geographic text into structured spatial data, which is crucial in fields such as urban planning, social media spatial analysis, and emergency response systems. Existing approaches predominantly model geocoding as a geographic grid classification task but struggle with the output space dimensionality explosion as the grid granularity increases. Furthermore, these methods generally overlook the inherent hierarchical structure of geographical texts and grids. In this paper, we propose a hierarchy-aware geocoding model based on cross-attention within the Seq2Seq framework, incorporating S2 geometry to model geocoding as a task for generating grid labels and predicting S2 tokens (labels of S2 grids) character-by-character. By incorporating a cross-attention mechanism into the decoder, the model dynamically perceives the address contexts at the hierarchical level that are most relevant to the current character prediction based on the input address text. Results show that the proposed model significantly outperforms previous approaches across multiple metrics, with a median and mean distance error of 41.46 m and 93.98 m, respectively. Furthermore, our method achieves superior results compared to others in regions with sparse data distribution, reducing the median and mean distance error by 16.27 m and 7.52 m, respectively, suggesting that our model has effectively mitigated the issue of insufficient learning in such regions.

Keywords: geocoding; cross-attention; sequence-to-sequence; geographic grid prediction

Citation: Liang, L.; Chang, Y.; Quan, Y.; Wang, C. A Hierarchy-Aware Geocoding Model Based on Cross-Attention within the Seq2Seq Framework. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 135. https://doi.org/10.3390/ijgi13040135

Academic Editors: Wolfgang Kainz and Dev Raj Paudyal

Received: 30 January 2024
Revised: 7 April 2024
Accepted: 15 April 2024
Published: 17 April 2024

1. Introduction

Geocoding involves resolving the location information described in natural language text to the corresponding points or regions on Earth [1,2]. With the popularization of the Internet and social media, there has been a dramatic surge in textual data associated with geographic information [3]. However, owing to privacy concerns, the availability of explicit geographic information, such as coordinates, on social media has gradually decreased. For example, Twitter discontinued explicit geographic information in 2019 and shifted to supporting implicit geographic information, such as POIs [4]. Therefore, geocoding has become an indispensable tool for mining big data on location-related social media and effectively extracting geographic information. It plays an important role in urban planning, disaster emergency response, the geospatial analysis of social media, and disease risk mapping [2,5,6].

Geocoding aims to parse unstructured text into structured spatial data, and its main output forms include geospatial coordinates, polygons, and entries into a geospatial database [1,2]. Previous geocoding systems relied mainly on external geospatial databases to match and sort text to be parsed with specific entries into the database [7–11]. However, owing to its high reliance on external knowledge bases, this form of geocoding faces

numerous obstacles in regions that lack standard geographic datasets or a GIS data infrastructure. In recent years, the application of end-to-end deep-learning models in geocoding tasks has gradually increased. This approach deepens the semantic understanding of address text and can directly predict geographical spatial labels from text with a straightforward workflow and low dependency on external databases such as gazetteers. It typically includes two primary methods: modeling as a regression task based on coordinate points or as a classification task based on regions such as grids or polygons [1–3,12–15]. Due to the challenges involved in directly learning the mapping between text and precise coordinates in coordinate-point-based tasks [15], some researchers prefer to model geocoding as a classification problem for predicting regions based on grids or polygons. Compared with approaches that use complex polygon structures for cities, states, and countries, geographic grids offer multi-level spatial representations without the need for external metadata, making them straightforward to understand and use. Therefore, this study focuses on grid-based classification methods for geocoding. This approach usually discretizes Earth's surface into a series of grids and predicts the grid category corresponding to the input text based on a classification model to indirectly obtain geographical coordinates [1,2,6,13–18]. Additionally, discrete global grid systems such as Geohash, H3, and S2 geometry play a crucial role in the spatial indexing, association, and geolocation of satellite and street-view imagery data [16,19–24]. Recently, research on the spatial alignment of multimodal data, such as text and images, has increased [25–27]. Parsing text into the corresponding geographic grids can provide a unified spatial foundation for these studies, offering extensive application prospects.

However, in fine-grained geocoding tasks with more detailed grid partitioning, the number of grid categories increases sharply. Taking the S2 grid as an example, when the average unit grid area is approximately 300 m^2, the global number of grids can reach 1649 billion. This leads to a serious dimensionality explosion in the output space, making it difficult for classification models to train and thus leading to a decrease in geocoding accuracy [15,28]. Moreover, most approaches generally overlook the inherent hierarchical structure characteristics of geographical grids and address text as well as the potential associations between them [29,30]. Unlike general texts, address texts often start with large-scale elements (e.g., countries and provinces) and are gradually refined into small-scale elements (e.g., specific buildings). Similarly, discrete global grid systems often recursively subdivide Earth from larger-scale grids into finer-scale grids. This structure implies that each sub-element (subgrid) represents varying hierarchies of geospatial information within a certain region and that text elements at different hierarchical levels often correspond to geographic grids of varying scales. Ignoring this feature may lead to the confusion of semantically similar but geographically different information [30].

To address these challenges, we propose a novel hierarchy-aware geocoding model based on cross attention (HAGM), which is a grid-based geocoding method within a sequence-to-sequence (Seq2Seq) framework. This study utilized S2 geometry (https://s2geometry.io/, accessed on 29 January 2024.) (a method for the hierarchical discretization of Earth's surface) to map continuous latitude and longitude coordinates onto discrete geographic grids (S2 cells) to represent geographic locations. Compared with traditional classification models, HAGM within the Seq2Seq framework treats each character of the grid labels as an independent unit to be predicted and sequentially outputs them character by character; thus, it effectively avoids the issue of dimensionality explosion in the output space. Moreover, the HAGM employs a cross-attention module and a residual connection module to effectively and comprehensively perceive the hierarchical structure of address texts and geographic grids, and establish a correspondence between input address text elements at different hierarchical levels and geographic grids of varying scales.

Our main contributions are the following:

(1) Our proposed method effectively avoids the influence of output space dimensionality explosion and performs well in fine-grained geocoding tasks.

(2) The HAGM dynamically focuses on the contextual information of varying geographic scales, thereby enhancing its perception of the hierarchical characteristics and potential semantic associations of address text and S2 tokens.

(3) We evaluate the impact of different grid division scales on the performance of the geocoding model and compare it with previous methods in terms of multiple evaluation metrics in order to provide a thorough analysis.

(4) The performance of HAGM in the sparse data distribution region is significantly improved compared with the traditional classification model; this indicates that HAGM can mitigate the effects of imbalanced data distribution and overcome the problem of insufficient model learning in areas of sparse data distribution to some extent.

2. Related Work

Early geocoding research primarily employed rule-based and traditional machine learning methods combined with heuristic strategies to establish direct mapping relationships between text and locations, obtaining candidate entries from address databases and then ranking and selecting them [7,9,31–33]. Although these methods have achieved satisfactory results, their frameworks are essentially based on indirectly predicting the corresponding geographic coordinates by ranking the similarities between the locations mentioned in the text and entries in a database, such as gazetteers. Owing to its high dependency on external databases, geocoding faces numerous obstacles in regions lacking standard geographic datasets or GIS infrastructure, leading to insufficient generalization capabilities [5,18,29]. Therefore, researchers have begun to apply end-to-end deep learning models to geocoding in order to directly predict associated geographic spatial labels based on input query texts. This is often modeled as a coordinate point-based regression task and a grid-based or region-based multi-classification task [12,13,15], with low dependence on external databases, such as gazetteers, and stronger generalization ability. Therefore, it is widely used in tasks such as event geocoding and Internet text geocoding [34,35].

In cases modeled as a coordinate-point-based regression task, researchers typically predict associated geographical coordinates directly from the input address text [34–38]. For example, Liu et al. [35] explored a method to estimate Twitter user locations using the textual data they generate on social media by utilizing a deep learning architecture constructed from stacked denoising autoencoders to directly predict the locations of users in terms of longitude and latitude. The results were comparable to those of the most advanced models at the time, demonstrating that this architecture is well-suited for geocoding tasks. Radford et al. [34] proposed an end-to-end probabilistic model for geocoding the text of event data, directly predicting the latitude and longitude of the locations mentioned or described in a natural language. They also compared their model-based solution with previous state-of-the-art open-source geocoding systems and extensively discussed the benefits of end-to-end geocoding based on their models. However, these methods are insufficient for extracting the semantic features of a text. Hence, Xu et al. [37] proposed a geo-semantic address model (GSAM) that supports various downstream tasks to deepen text features. Building on this, they incorporated three fully connected layers as hidden layers for the address location prediction task and added a final linear layer with two neurons to directly predict the coordinates (latitude and longitude). However, this regression-based approach to directly predicting geographic coordinates can cause issues with the continuity and infinity of the output space, resulting in learning difficulties for the model and often leading to serious degradation of the model performance owing to data quality issues.

Consequently, researchers tend to model the geocoding problem as a classification task [1,2,6,13,14,16–18] in which Earth's surface is discretized into a series of grids, and the model directly predicts the specific grid category corresponding to a geospatial label based on the input address description. For example, DeLozier et al. [18] computed the geographic profile of each word using local spatial statistics on a set of geo-referenced language models with a machine learning-based classification model for toponym

resolution, significantly outperforming other advanced toponym resolvers of the time; however, previous methods have mostly focused solely on lexical features, excluding other feature spaces. To address this issue, Gritta et al. [6] introduced the Map Vector (MapVec), a sparse representation that simulates the geographic distribution of location mentions. They proposed the CamCoder model, which integrates three lexical feature vectors and one sparse geographic vector, feeding them into a dense layer for final region classification, and achieved state-of-the-art results on three different datasets. Cardoso et al. [14] further attempted to optimize geocoding by combining two outputs of classification and regression. They adopted the grid partitioning method based on hierarchical equal area isolatitude pixelization (HEALPix) and utilized context-aware word embeddings such as ELMo and BERT to transform the input text. The transformed text was then fed into a bidirectional LSTM unit-based neural architecture to derive the grid classification results. Additionally, they obtained coordinate outputs using class probability vectors alongside centroid coordinate matrices, and the model was trained by a comprehensive classification and regression loss function, surpassing prior studies. To achieve a balance between generalization and accuracy, Kulkarni et al. [1] introduced a CNN-based multi-level geocoder MLG. It employs multi-level S2 cells as outputs for the multi-head feature encoding model, integrates losses at multiple levels, and predicts cells at each level simultaneously, achieving better performance than CamCoder. However, classification models commonly encounter the issue of output space dimensionality explosions in geocoding tasks. Although researchers have attempted optimization using strategies such as hierarchical nested grids [29] and multitask joint prediction [28], these have shown only slight improvements compared with previous studies and have not fundamentally resolved the high complexity of the output space. In particular, high-precision, fine-grained geocoding tasks with more detailed grid partitioning suffer from severe dimensionality explosions and underperform.

The Seq2Seq framework offers a novel perspective for geocoding tasks. The Seq2Seq framework is a deep learning model architecture used for handling sequence data and has demonstrated good performance in various natural language processing tasks such as machine translation and text summarization [39–47]. It typically comprises two components: an encoder and a decoder. The encoder is generally responsible for converting an input sequence (such as a text sequence or time series) into a fixed-length vector. The decoder receives the vector representation output from the encoder and gradually generates the target sequence. Given the limited character categories used in the grid label sequence, we posit that the character-by-character prediction of the Seq2Seq model effectively avoids the problem of dimensionality explosion, thereby enhancing geocoding performance. Qian et al. [48] combined a GeoSOT grid-division system with a sequence-to-sequence framework to design a coarse-to-fine model to solve text geolocation problems. However, the Z-order curve used by GeoSOT [49] suffers from a local order mutation at its zigzag corners. Huang et al. [15] used S2 geometry, indexed by the Hilbert curve with stronger local order preservation, to represent geographic locations. They attempted to treat multi-level grid label encoding sequences as collections of individual characters for independent classification, which enhanced computational efficiency through parallel processing, but overlooked the potential correlations between characters. Moreover, these methods are associated with limitations in capturing the correspondence between address elements at different hierarchical levels and grids of varying scales, leading to a decrease in localization accuracy and precision. Specifically, both the input address text and output grid labels have a natural hierarchical structure. Geocoding models should fully utilize multiscale information by focusing on the address elements of the corresponding hierarchical level when predicting characters for grids of different scales. Failure to capture this hierarchical relationship can result in the confusion of semantically similar but geographically different information, severely affecting geocoding accuracy [17,30]. Therefore, overcoming this limitation is crucial for enhancing model performance.

In this study, we used the S2 geometry to discretize Earth's surface into grids and proposed a hierarchy-aware geocoding model based on cross-attention within a sequence-to-sequence framework. This model effectively and comprehensively perceived the hierarchical structure of the address text and geographic grid through a cross-attention mechanism and a residual connection module. In each prediction step, the model dynamically perceives and focuses on different hierarchical levels of address context information, thereby establishing a correspondence between the address text elements and geographic grids. Additionally, we conducted comprehensive evaluations of the model at various grid division scales in order to provide a deeper understanding and analysis of grid-based geocoding methods.

3. Methodology and Model

We propose a grid-based hierarchy-aware geocoding model (HAGM) that incorporates a cross-attention mechanism within the Seq2Seq framework, aimed at implementing geocoding by learning the mapping relationship between the address text and the corresponding geographic grid labels. Using S2 geometry, we first mapped the latitude and longitude coordinates onto discrete geographical grids (S2 cells), with S2 tokens (label sequences of S2 cells) serving as labels for the geographical location to be predicted. Subsequently, we treated each character in S2 tokens as an independent unit within the Seq2Seq framework and predicted the target S2 tokens character-by-character based on the input address text. As shown in Figure 1b, this character-by-character output approach confines the prediction space of each step to a 16-character category (10 Arabic numerals 0–9 and six English letters a–f), effectively avoiding the computational challenges caused by dimension explosion. Additionally, the HAGM utilizes a cross-attention module to dynamically focus on the input address elements of the most relevant hierarchical level during each step of the decoder, accurately capturing contextual information closely related to the corresponding level of geographic grids. Furthermore, it retains the original global address context through the residual connection, thereby effectively and comprehensively perceiving the hierarchical structure of the address text and the geographic grid and establishing the correspondence between address elements at different hierarchical levels and grids of varying scales. This approach ensures that the prediction of each character relies on the previous characters and the context information of the relevant address text elements. Finally, the HAGM makes an overall prediction by optimizing the total loss of all character calculations. Furthermore, following prior research [29], we use the center point coordinates of the S2 cell corresponding to the predicted S2 tokens as the final predicted geographic location. The overall methodology is illustrated in Figure 1, with a more detailed description of the S2 geometry and specific model structure in Sections 3.1 and 3.2 of this chapter.

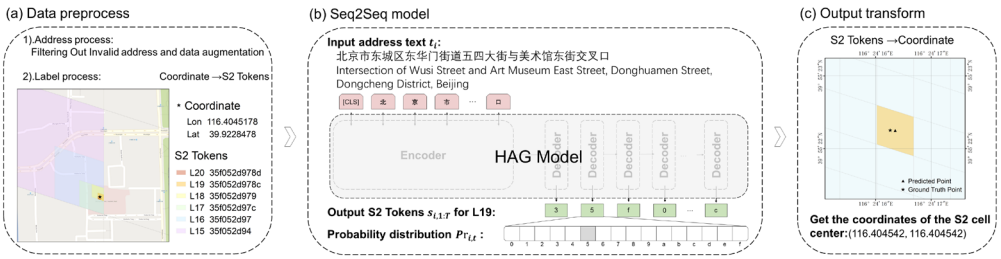

Figure 1. Overall method framework.

Finally, this study comprehensively evaluated the model using a range of assessment metrics and compared it with previous mainstream models. These aspects are discussed in Section 4.

3.1. S2 Geometry and Grid Division

In this study, we used S2 geometry to represent geographic locations. It is a hierarchical discretization method for Earth's surface that recursively divides Earth's surface into four quadrants using the Hilbert curve, enabling a natural multi-level spatial representation [29,50,51]. The series of geographic grids obtained by partitioning Earth's surface using the S2 geometry are called S2 cells [50]. Table 1 shows information on the various levels of S2 cells. Each S2 cell is uniquely identified using a 64-bit S2 cell ID. The longer the effective bits of the cell ID, the higher the corresponding level, and the finer the grid resolution, the smaller the geographic areas. The S2 cells were sequentially numbered along a specific space-filling curve, ensuring that cells with adjacent S2 cell IDs were also spatially adjacent (http://s2geometry.io/devguide/s2cell_hierarchy.html/, accessed on 29 January 2024.). S2 tokens are hexadecimal string representations of the S2 cell IDs. The front characters of the S2 token sequence represent broader geographical scales, whereas the back characters often indicate finer scales, thus achieving a multi-level description of the geographic space.

Table 1. Granularity of S2 cells at different levels.

S2 Level	Number of Cells	Avg Area
15	6B	79,172.67 m^2
16	25B	19,793.17 m^2
17	103B	4948.29 m^2
18	412B	1237.07 m^2
19	1649B	309.27 m^2
20	7T	77.32 m^2

We conducted experiments across various S2 levels ranging from 15 to 20, evaluated the model using comprehensive metrics, and compared it with prior mainstream models.

3.2. The Proposed Model Architecture

The HAGM consists of an encoder based on a pretrained language model and a decoder enhanced by a cross-attention mechanism. The encoder captures deep semantic features in the input text based on the pretrained RoBERTa language model, producing high-quality semantic representations. The decoder combines a recurrent neural network model with a cross-attention mechanism to perceive the hierarchical structure within the address text dynamically and accurately capture contextual information closely associated with the corresponding scale of the geographic grid. Additionally, the model adopts residual connections to preserve the original global context information and applies layer normalization to optimize performance. Finally, the processed combined information is transformed into a probability distribution matching the size of the target vocabulary through a linear classification layer, enabling character-by-character prediction of the S2 tokens. The model architecture is shown in Figure 2, with detailed structures of the encoder and decoder described in Sections 3.2.1 and 3.2.2, respectively.

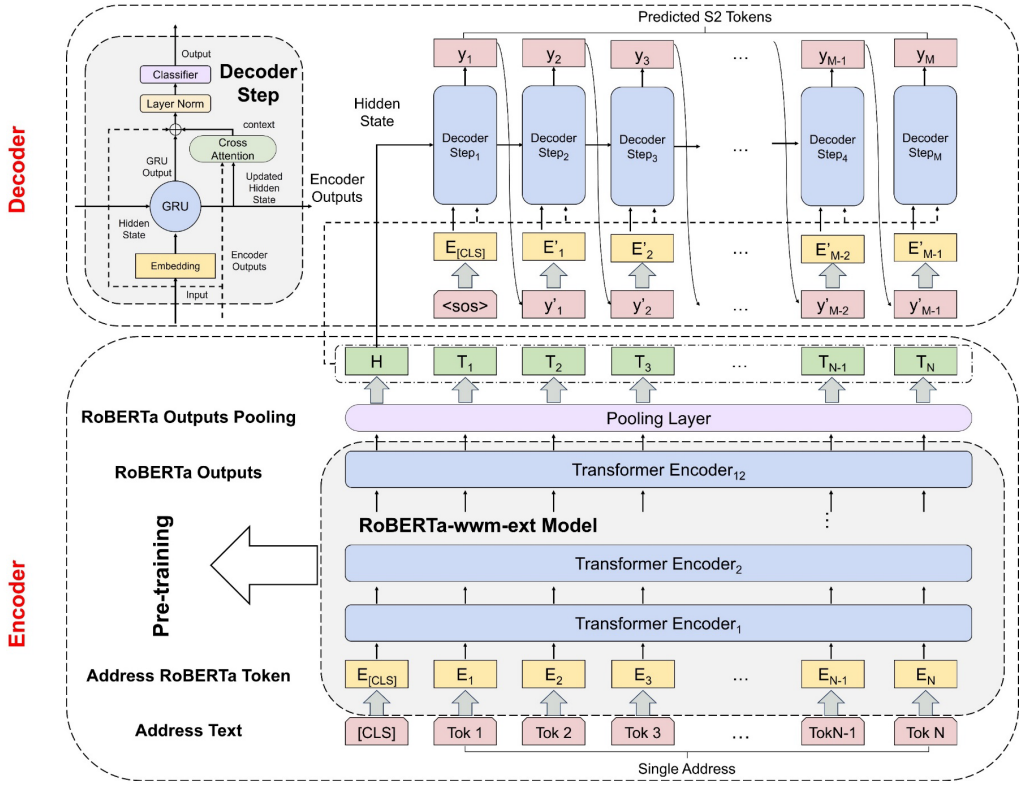

Figure 2. HAGM architecture.

3.2.1. Encoder

For the input address text, we designed a transformer-based encoder to capture rich semantic information within the input text and extract high-quality semantic representations. It was stacked with 12 transformer modules, each with hidden layer dimensions of 768. Every module integrates a multiheaded self-attention mechanism and a feed-forward neural network and applies layer normalization techniques. Using this structure, the model can perform deep feature extraction and semantic information capture from the input text, thereby providing a powerful feature extractor for downstream tasks. To better adapt to the characteristics of Chinese address text, we initialized the transformer-based encoder using the pretrained RoBERTa-chinese-wwm-ext model parameters [52,53], which utilizes the whole-word masking (wwm) strategy and is optimized on Chinese data. Furthermore, a fully connected layer was added after the output of the transformer-based encoder, projecting the feature vector dimensions from 768 to 512, which served as the input for the decoder.

Specifically, for an input query text, we first tokenized the text $T_{i,0:l-1}$ into subword-level input representations $t_{i,0:l-1}$ which were then fed into the pretrained RoBERTa model in order to obtain a deep representation of RoBERTa denoted as $R_{i:0:l-1}$.

$$R_{i,0:l-1} = RoBERTa\left(t_{i,0:l-1}\right) \quad (1)$$

Subsequently, we appended a fully connected layer to the end of the RoBERTa model for pooling, which compressed the dimensionality of the output from 768 to 512, thereby producing the final encoder output, denoted as E_i.

$$E_{i,0:l-1} = W \cdot R_{i,0:l-1} + b \quad (2)$$

This helped the model to better compress, integrate, and transform the information from the input address text in order to adapt to the requirements of the decoder, optimize computational efficiency, and meet the needs of the downstream S2 encoding prediction task.

3.2.2. Cross-Attention-Enhanced Decoder

As mentioned in Section 2, previous studies [15,29] have often overlooked the hierarchical characteristics of the address text and S2 tokens. To address this, we employed a cross-attention-enhanced recurrent neural network as the decoder, and a specific type of recurrent neural network was chosen as the gated recurrent unit (GRU) [40]. With its gating mechanism, the GRU can effectively alleviate the problem of gradient disappearance and maintain long-term dependencies when dealing with long-term time series data. Thus, it performs well at capturing details and context relationships within address texts, surpassing traditional recurrent neural networks.

Specifically, the initial hidden state of the decoder is defined as $hidden_{i,0} = E_{i,0}$, where $E_{i,0}$ refers to the vector representation of the first token "[CLS]" output by the encoder. We use the token "<sos>" as the initial input, denoted as $input_{i,0}$, and the update formula of GRU is:

$$r_{i,t}, hidden_{i,t} = GRU\left(input_{i,t}, hidden_{i,t-1}\right) \tag{3}$$

To effectively capture the hierarchical relationship between address text elements and S2 tokens, we introduced a cross-attention mechanism into the decoder. While predicting each character by calculating the cross-attention scores between the current hidden state of the output sequence and various parts of the deep feature representations of the input address text sequence, the decoder dynamically weights the input address text, thus facilitating effective interaction between different modalities. This allows the model to focus dynamically on the most relevant hierarchical levels of address text elements at each character prediction step, thereby accurately capturing the contextual information closely associated with the corresponding scale of the geographic grid.

Specifically, for the current hidden state $hidden_t$, we compute the cross-attention weights $\alpha_{i,j,t}$ and the weighted context vector $context_{i,t}$ with the semantic vectors E_i output by the encoder:

$$\alpha_{i,j,t} = \frac{\exp\left(f\left(hidden_{i,t}, E_{i,j}\right)\right)}{\sum_{k=0}^{l} \exp\left(f\left(hidden_{i,t}, E_{i,k}\right)\right)} \tag{4}$$

$$context_{i,t} = \sum_{j=0}^{l} \alpha_{i,j,t} \cdot E_{i,j} \tag{5}$$

Given the characteristics of S2 tokens, the front characters of the sequence represent broader geographical scales, whereas the back characters tend to indicate finer scales. This makes it possible to precisely focus on the address element information of the corresponding hierarchical level when predicting characters at different positions of the S2 tokens. This mechanism enables effective information interaction between the input address text and output grid label sequence. Additionally, we effectively preserved the original global context information through residual connections, promoted efficient information transfer, and enhanced the model learning capability. By balancing local and global information, our model efficiently and comprehensively perceives the hierarchical structure of address texts and geographic grids, establishing a correspondence between address text elements at different hierarchical levels and geographic grids of varying scales.

$$context_{i,t} = context_{i,t} + E_{i,j} \tag{6}$$

Ultimately, the decoder needs to integrate the output of the GRU with the contextual information based on attentional weighting and transform it through a linear layer into a probability distribution equal to the size of the target vocabulary, thereby completing the character-by-character prediction of the S2 tokens.

$$combined_{info_{i,t}} = Concatenate\left(r_{i,t}, context_{\ i,t}\right) \tag{7}$$

$$\mathrm{Pr}_{i,t} = Softmax\left(FC\left(LayerNorm\left(combined_{info}{}_{i,t}\right)\right)\right) \quad (8)$$

Specifically, $\mathrm{Pr}_{i,t}$ represents the probability distribution of the output classified as different S2 encoding characters c_n at time step t, where $n \in \{0,1,\ldots 15\}$ (with c_n including the 6 English letters a–f and 10 Arabic numerals 0–9), i.e., $\mathrm{Pr}_{i,t}$ is a 16-dimensional probability vector. We define the predicted character of the final output based on $\mathrm{Pr}_{i,t}$ as $s_{i,t}$ and simultaneously use it as the input for the next time step, denoted as $input_{i,t+1}$. This process is repeated until a full S2 token sequence is generated, with the resulting sequence being represented as $S_{i,1:T} = \{s_{i,1}, s_{i,2}, \ldots, s_{i,T}\}$ and the center of the S2 cell being used as the final predicted coordinate.

To ensure the robustness of the model and accelerate the training process, we introduced layer normalization (LN) [54] before the final classification layer. The LN can normalize the input of each layer, thereby smoothing the flow of information and enhancing the robustness of the model.

3.3. Evaluation Metrics

We utilized four commonly used geocoding metrics to evaluate the model comprehensively: accuracy [3] (also known as accuracy @N km), mean distance error, median distance error, and area under the curve (AUC) for the error curve [55]. In this study, we first used the Haversine formula (a well-known method for calculating the geodetic distance between a pair of latitude and longitude points on an ellipsoidal Earth model) to calculate the great-circle distances between the predicted coordinates (x_pred, y_pred) and ground-truth coordinates (x_label, y_label), and then calculated each evaluation metric based on the obtained error distances.

Accuracy @N km measures the percentage of predicted locations that are less than N km from the true location. A higher percentage indicates that most predicted locations are within the allowable error threshold from the actual location, indicating that the model has higher precision in predicting geographical locations. Given the scope of this study, we set N to 0.05, which is equivalent to 50 m, because an error within 50 m can be considered relatively accurate. Although this metric is direct and straightforward, it disregards all errors exceeding 50 m.

Mean Distance Error measures the average distance between all predicted and true locations of the target address, and lower error values are preferred [14]. It is calculated by dividing the sum of all the geocoding errors by their total number, revealing the overall performance of the geocoder and the general error trend. However, it is extremely sensitive to outliers, because it treats all errors as equivalent.

Median Distance Error measures the median distance between all the predicted locations and the true locations of the target address. Lower values are desired, signifying a greater alignment of the predictions with the true location. Compared to the mean distance error, the median distance error provides a more robust assessment of the performance of the geocoder because it is not affected by extreme errors.

AUC represents the area under the discrete curve of the sorted error distances, which serves as a significant comprehensive metric [55] because it captures the overall distribution of errors and is not affected by outliers. The larger the AUC value, the more stable the performance of the geocoding model. In our study, the AUC was calculated using the logarithm of these distances, which shifted the focus of the metric towards smaller error distances when comparing models and reduced the importance of larger errors.

A versatile geocoding model should comprehensively consider all metrics in order to maximize performance [6]. We further explored the performances and trends of the various models at different S2 levels.

4. Experiments and Results

4.1. Study Area and Dataset

We collected addresses from Dongcheng District, Beijing as the dataset for this study. Located in central Beijing, Dongcheng District covers an area of approximately 41.84 km^2 and encompasses 17 streets and 177 communities. At the S2 level 20, this area contained 644,334 S2 cells. The original address dataset comprised 64,025 entries, covering various types of addresses, such as restaurants and shopping malls. Each entry includes detailed field information, such as place name, address description, longitude, latitude, and administrative divisions at various levels. We selected address descriptions containing rich geographical information as our input data. However, owing to irregularities in data entry and the variety of formats used, the quality and validity of the data vary; therefore, appropriate data processing is required.

To ensure the accuracy and consistency of the data, we conducted targeted preprocessing, which included: (1) identifying and removing duplicate and empty addresses in the dataset; (2) correcting invalid data, such as addresses with internal repetition or those containing special characters, full-width characters, or half-width characters; (3) combining the administrative division fields and name fields so as to randomly adjust address descriptions through supplementation, deletion, and concatenation, aiming to simulate query texts in real-world scenarios, and thereby increasing the diversity and practicality of address data. After preprocessing, we obtained 59,717 valid addresses, of which the longest address contained 82 tokens, and the average number of tokens for all addresses was 28.21. The text types considered in this study included completely hierarchically structured address descriptions and address descriptions with missing hierarchical elements. Some examples of address data are shown in Table 2, in which some data lack elements such as cities, districts, streets, or roads, indicating that they are not standard hierarchical structured addresses.

Table 2. Data examples.

Address (Chinese)	Address (English)	S2 Tokens			
		Level 15	Level 16	…	Level 20
北京市东城区东华门街道南池子大街 85 号德景盛	Dejing Sheng Building, No. 85 Nanchizi Street, Donghuamen Subdistrict, Dongcheng District, Beijing	35f052bfc	35f052bff	…	5f052bfee9
东城区和平里街道和平里七区 16 号楼 530 室邦利生活	Bangli Life, Unit 530, Building 16, Hepingli Seventh Block, Hepingli Subdistrict, Dongcheng District, Beijing	35f05354c	35f05354d	…	35f05354dd7
北京市龙潭街道广渠门内大街安化北里 1 号院	No. 1 Court, Anhua North Alley, Guangqumen Inner Street, Longtan Subdistrict, Beijing	35f1ad5ac	35f1ad5ab	…	35f1ad5aa7b
北京市东城区朝内小区 7 号楼二单元 701 室	Unit 2, Room 701, Building 7, Chaonei Community, Dongcheng District, Beijing	35f1ad2a4	35f1ad2a3	…	35f1ad2a39b

We leveraged the S2 geometry to convert the geographic coordinates of each address description into the corresponding S2 tokens, serving as the ground truth for the addresses. Specifically, we utilized the Python package s2sphere to calculate the S2 cell grid labels (S2 Tokens) for each address. First, we converted the latitude and longitude coordinates of each address from degrees to radians. Subsequently, based on the converted radians, we identified the S2 cell at different levels that contained the given geographic location and calculated its corresponding grid ID. Finally, we converted the S2 cell ID into the corresponding S2 tokens and used them as the ground-truth label for the address. Thus, we completed the conversion of the output space, which satisfied the modeling requirements of the sequence-to-sequence analysis. Finally, we split the preprocessed 59,717

labeled valid address data in a 9:1 ratio. The training set contained 90% of the data and was dedicated to model training. The remaining 10% of the data served as the test set to evaluate the model's performance.

4.2. Training and Experimental Setup

4.2.1. Training Details

During the training process, the input of our model was the preprocessed address description, the output was the predicted sequence of S2 tokens, and the corresponding truth labels were the sequences of S2 tokens previously computed from the latitude and longitude labels of the address text. The predicted output of each sequence character was a 16-dimensional vector of probability distributions, as shown in Figure 1b. We selected the character with the highest probability at each prediction step as the final output for that time step and used it as the input of the next time step. For each time step, we computed the cross-entropy loss between the probability distributions of the characters predicted by the model and the real characters, summing the losses of all time steps to obtain the total loss. By minimizing the total loss, we can optimize the model parameters to better understand the relationship between the input and output sequences. Additionally, during the training, we employed a teacher-forcing strategy. This strategy involved using true sequence characters as inputs for the next time step with a certain probability, rather than relying solely on the model's predicted results. This approach helped increase the robustness of the sequence-to-sequence model and accelerated the convergence process of the model. The total loss was calculated using the following equation:

$$\mathrm{Loss}_{total} = \sum_{t=1}^{T} \mathrm{CrossEntropyLoss}(\mathrm{Pr}_t, y_t) \quad (9)$$

Our model was trained on a single NVIDIA GeForce RTX 3090 GPU with a memory of 24 GB using the PyTorch framework. To better handle the difference in structural complexity between the encoder and decoder, we set different learning rates: the learning rate for the encoder was set to 1×10^{-5}, and the learning rate for the decoder was set to 7×10^{-4}. After the initial warm-up phase, the learning rate decayed exponentially with an increase in batches, which helped stabilize the training process of the model. Throughout the training process, we used the Adam optimizer for stable parameter updates, processing 64 samples per training batch, and training the model for a total of 200 epochs. Adam is an algorithm used for optimizing gradient descent. This algorithm is especially suitable for dealing with sparse gradients and adaptively adjusting the learning rate and can effectively improve the efficiency of deep learning model training [56]. To improve the generalizability of the model and prevent overfitting, we introduced dropout technology and set it to 0.5.

4.2.2. Experimental Setup

To compare the performance of our model with other geocoding models, we selected the SLG [29] (single-level geocoding), MLG [29] (multi-level geocoding), and MLSG [14] (multi-loss geocoding) models as baselines to evaluate the strengths and improvements of our proposed model in fine-grained geocoding tasks. These models are representative of different processing strategies and design ideas for geocoding tasks in recent years. Among them, the SLG model is the most basic grid-based classification geocoding method. The MLG model mitigates the problem of growing output spatial dimensionality by combining multilevel grids, and the MLSG model demonstrates another way to cope with the spatial complexity of high-dimensional outputs by combining the classification loss with the coordinate regression loss. All of these methods employed RoBERTa to obtain feature representations from the input address and aimed to predict the corresponding grid label and coordinates under the S2 partition as their shared objective in order to ensure consistent evaluation.

Moreover, to evaluate our model comprehensively, we categorized the study area into sparse regions (less than 2), dense regions (more than 10), and regular regions (between 2 and 10) based on the number of address data entries within the unit area. We evaluated the performance metrics for each of these regions as well as the overall region to verify the consistency and accuracy of the model for various geographical data distributions. We also assessed the accuracy and generalization of the model at all levels, from 15 to 20, to explore its performance at different spatial granularities, thereby determining the optimal scale for geocoding. Additionally, we explored the model performance in fine-grained geocoding tasks by setting N in the Accuracy@N km metric to 10, 25, and 50.

To explore the effectiveness of the model and the contribution of each component, we conducted ablation experiments on key modules, including the cross-attention mechanism and layer normalization. Furthermore, to enable meaningful comparisons with the method in [48], we focused on core module differences, given our adoption of more advanced backbone and grid partitioning techniques, along with the integration of residual connections and layer normalization modules. Specifically, we replaced the HGAM's cross-attention module with the self-attention module used in [40] to achieve a fair comparison. In this way, we can analyze the respective contributions of each module to the model performance, allowing for a more comprehensive and objective evaluation of our model.

4.3. Results

We evaluated the performances of SLG, MLG, MLSG, and our model across different density distribution areas and various S2 levels for all metrics. All of the results of the comparison models were based on our own implementation, and training and testing were conducted under the same environment and dataset to ensure the fairness of the experiments.

4.3.1. Overall Trends

As shown in Tables 3 and 5, our model demonstrated a significant advantage across all evaluation metrics compared to SLG, MLG, and MLSG. The AUC metric of our model reached its highest value at 0.59, surpassing the values achieved by SLG, MLG, and MLSG by 4, 5, and 6 percentage points, respectively, implying that our model had an excellent overall error distribution that tended towards lower error values. The median distance error of our model was as low as 41.46 m, which is at least 6 m better than that of the other models. For the more stringent metric, Accuracy@50m, our model scored up to 0.56, which is an improvement of at least 4 percentage points compared to other models, suggesting that most of the predictions of our model are quite accurate. Moreover, the mean distance error metric of our model was 93.98 m, outperforming SLG, MLG, and MLSG by approximately 31 m, 26 m, and 8 m, respectively. This implies that our model maintains good predictive quality in most scenarios, demonstrating its resilience to noise and robustness. As shown in Figure 3, the predicted results of our model closely matched the actual distribution in most regions with smaller distance errors.

Table 3. Performance comparison of SLG, MLG, MLSG, and our model at their optimal levels.

Model	AUC	Accuracy@50m	Median	Mean	Best Level
SLG	0.55	0.50	50.49	124.95	17
MLG	0.54	0.52	47.20	120.19	18
MLSG	0.53	0.50	49.18	101.90	17
HAGM	0.59	0.56	41.46	93.98	20

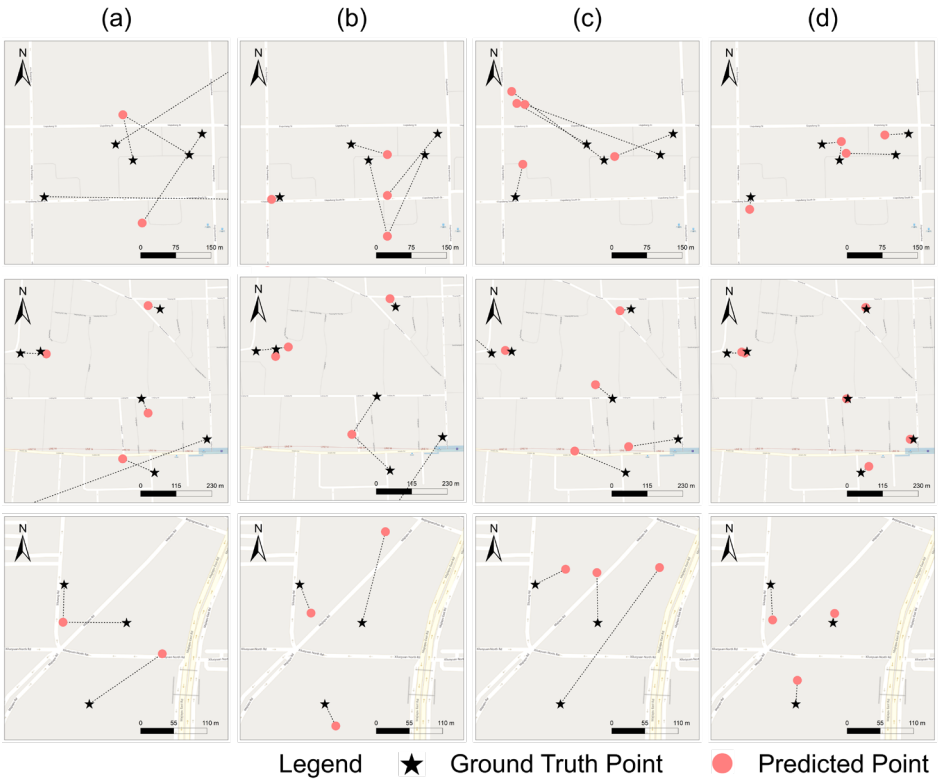

Figure 3. Examples of geocoding results visualization for (**a**) SLG, (**b**) MLG, (**c**) MLSG, and (**d**) HAGM (ours) at their optimal levels.

Furthermore, our HAGM model achieved the best accuracy when the S2 level was set to 20, whereas the other models peaked at S2 levels of 17 or 18. As shown in Table 5, at L20, our model surpassed the baseline models by at least 3 percentage points on the AUC and outperformed comparative models by a minimum of 9 percentage points on the Accuracy@50m dataset. Meanwhile, the median and mean distance errors of our model were reduced by at least 13.72 m and 30.42 m, respectively. As shown in Table 4, our model achieved the best results for the Accuracy@10m, Accuracy@25m, and Accuracy@50m metrics. This suggests that when the output space features more detailed and refined grid partitioning, the HAGM can successfully learn more granular spatial information than the other models, making it better suited for learning fine-grained geocoding tasks.

Table 4. Fine-grained geocoding evaluation results of SLG, MLG, MLSG, and our model.

Model	Accuracy@50m	Accuracy@50m	Accuracy@50m
SLG	0.16	0.27	0.50
MLG	0.24	0.33	0.52
MLSG	0.19	0.31	0.50
HAGM	0.28	0.38	0.56

In summary, our model demonstrates a more comprehensive and balanced performance with significant advantages, which is in line with the standards of a versatile geocoder. Table 5 and Figure 4 provide a detailed presentation of the performance of the model at different S2 levels and in regions with varying data distributions; a deeper analysis and discussion will be carried out in the subsequent sections.

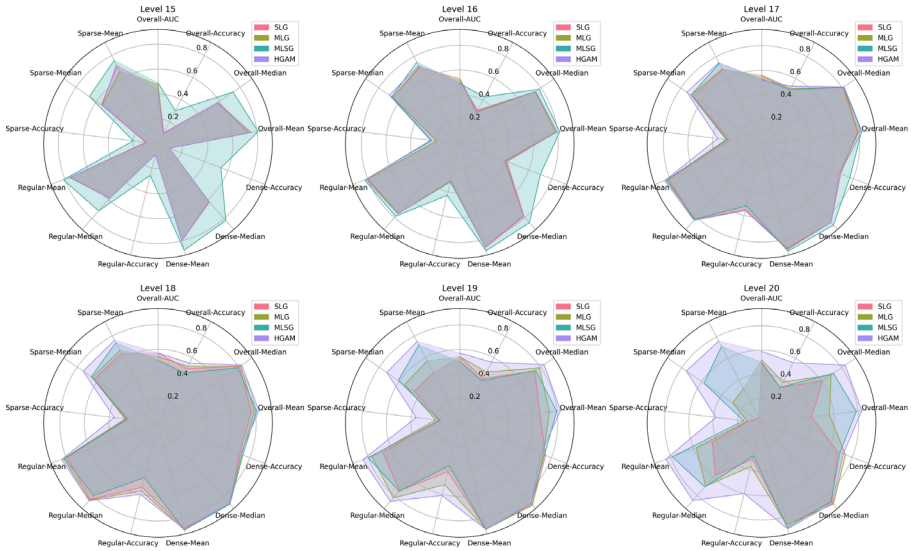

Figure 4. Visualization of performance comparison between SLG, MLG, MLSG, and HAGM. The evaluation results presented in the graph are, in clockwise order, overall, dense area, regular area, and sparse area. We have normalized the mean and median distance errors for intuitive comparison. A larger value on the radar chart indicates better performance on that metric.

Table 5. Performance comparison between SLG, MLG [1], MLSG [14], and our HAGM on the same dataset across different S2 levels and area densities.

Level	Region	AUC				Accuracy@50m				Median				Mean			
		SLG	MLG	MLSG	HAGM	SLG	MLG	MLSG	HAGM	SLG	MLG	MLSG	HAGM	SLG	MLG	MLSG	HAGM
15	Overall	0.47	0.48	0.49	0.43	0.10	0.09	0.30	0.09	115.82	117.21	72.52	117.08	148.85	160.64	108.76	146.26
	Dense					0.10	0.10	0.54	0.10	104.95	105.94	46.27	106.42	114.75	115.66	61.93	116.26
	Regular					0.09	0.09	0.26	0.09	115.96	117.07	76.67	117.29	144.84	147.92	107.06	143.16
	Sparse					0.10	0.09	0.20	0.09	125.49	128.57	91.30	127.76	180.52	214.73	145.97	173.41
16	Overall	0.53	0.53	0.49	0.51	0.30	0.29	0.43	0.31	68.36	68.85	57.74	67.33	124.64	125.07	102.99	117.45
	Dense					0.41	0.39	0.70	0.40	55.73	58.47	36.76	58.93	70.20	72.85	51.43	72.53
	Regular					0.31	0.31	0.43	0.32	65.21	65.54	56.55	64.94	112.37	115.89	97.29	104.81
	Sparse					0.19	0.19	0.23	0.24	93.11	92.94	90.33	83.49	184.93	178.59	150.31	171.35
17	Overall	0.55	0.56	0.53	0.52	0.50	0.51	0.50	0.53	50.49	49.28	49.18	47.35	124.95	118.80	101.90	98.77
	Dense					0.69	0.68	0.77	0.68	36.57	36.57	27.88	36.63	65.70	60.62	47.17	56.96
	Regular					0.55	0.56	0.52	0.56	44.63	44.04	46.47	43.31	108.55	100.47	93.33	85.29
	Sparse					0.26	0.29	0.28	0.36	88.71	82.90	82.54	70.12	195.59	191.87	156.28	151.78
18	Overall	0.57	0.54	0.51	0.57	0.50	0.52	0.46	0.55	50.54	47.20	55.56	43.71	138.36	120.19	108.31	96.54
	Dense					0.71	0.73	0.74	0.72	27.81	27.93	27.88	29.50	57.94	48.72	47.49	53.36
	Regular					0.54	0.57	0.46	0.60	42.94	40.83	55.14	38.63	121.59	99.18	105.02	88.35
	Sparse					0.26	0.27	0.25	0.36	103.01	96.21	93.48	69.86	225.17	207.43	158.41	141.80
19	Overall	0.54	0.54	0.50	0.57	0.41	0.47	0.39	0.56	68.77	55.30	66.15	41.58	212.94	158.11	117.47	95.54
	Dense					0.75	0.73	0.74	0.71	25.16	23.89	29.22	28.69	52.59	50.63	49.85	49.20
	Regular					0.41	0.52	0.36	0.61	70.01	47.03	68.74	36.00	202.72	137.13	114.89	83.23
	Sparse					0.16	0.20	0.17	0.36	154.87	126.95	108.40	74.83	347.60	271.79	171.40	149.94
20	Overall	0.49	0.52	0.50	0.59	0.33	0.38	0.33	0.56	108.32	83.10	76.33	41.46	379.84	277.89	124.40	93.98
	Dense					0.67	0.74	0.69	0.72	30.94	25.25	33.71	28.07	68.56	69.25	53.69	47.12
	Regular					0.31	0.37	0.28	0.60	118.94	83.29	80.71	36.46	366.49	269.65	124.06	85.27
	Sparse					0.11	0.12	0.15	0.38	273.02	203.55	118.38	66.27	630.53	444.74	176.89	142.79

4.3.2. Performance across Different S2 Levels

To explore the performance of the models at various spatial partitioning scales, we evaluated the accuracy of each model at the S2 levels 15–20. As shown in Figure 4 and Table 5, the accuracy of SLG, MLG, and MLSG initially showed an increasing trend as the S2 level increased but started to decline after reaching an optimum between L17 and L18. Meanwhile, HAGM demonstrated weaker performance at L15 and L16, but steadily improved as the S2 level increased, surpassing other models and peaking at L20. This

suggests that the HAGM can learn granular spatial information more effectively and handle high-resolution geospatial data.

As the S2 level increases, the granularity of the output space partitioning becomes even more refined, leading to a sharp increase in the number of S2 cells within the study area; thus, traditional classification models such as SLG, MLG, and MLSG face the problem of dimensionality explosion in the output space, which leads to a decrease in accuracy. Although MLG and MLSG have attempted to mitigate this issue by fusing the results from multiple levels or adopting weighted predictions, their performance still suffers at finer scales, such as L19 and L20. By contrast, HAGM adopts a character-by-character prediction strategy in which each character's prediction involves only a 16-dimensional output space, and, thus, effectively avoids this problem and successfully gathers knowledge at finer scales.

4.3.3. Performance across Different Area Densities

To explore the performance of the models across varying data distribution regions, we assessed the precision of each model in the sparse, regular, dense, and overall regions. As shown in Table 5 and Figure 4, all of the models exhibited substantially lower performance metrics in sparse and regular regions than in dense regions. This difference can be attributed to the higher data density in denser regions, which provides the model with more opportunities to capture geospatial patterns. By contrast, sparse regions have a more scattered data distribution, causing the model to be easily influenced by dense regions and thereby leading to bias, which is also consistent with the view expressed in [29].

Specifically, as shown in Figure 4, as the level increased beyond the respective optimal S2 level, the performance indicators in the dense regions for the baseline models did not decline severely. However, at the same time, there is a significant decline in performance in sparse and regular regions, as observed in the performance of MLG and MLSG at L18, L19, and L20 in sparse and regular regions, which ultimately impacts the model's overall performance. This suggests that the underperformance of geocoders based on grid prediction methods in high-precision geocoding tasks may stem from their inadequacy when learning from sparse regions.

In contrast, our model shows significant advantages in both sparse and regular regions while maintaining good performance in dense regions. Particularly at L20, the Accuracy@50m of our model achieves 0.38 in sparse regions, an improvement of at least 9 percentage points compared to other models, while the median and mean metrics reach 66.27 m and 142.79 m, respectively, a reduction of 16.27 m and 7.52 m. These figures indicate that our model can effectively learn geospatial distribution patterns even in sparse regions with strong generalization and stability across different data distributions. This advantage may be related to the strategy of our model, which tends to learn a pattern of mapping from address elements of different hierarchical levels to the corresponding S2 tokens characters. Therefore, even in sparse regions, our model can apply the knowledge of spatial hierarchies learned from other regions, partially reducing the impact of the data distribution density and achieving relatively good accuracy.

4.3.4. Performance across Various Address Types

To explore the performance of the models across different data types, we evaluated each model for complete and incomplete address descriptions with missing elements. The results indicate that our model outperformed the other models for both address types. In particular, as shown in Table 6, our model achieved the best median and mean distance errors in incomplete address descriptions, reducing at least 6.30 m and 27.15 m compared to the baseline model. Moreover, it showed a significant advantage, with a 4-percentage-point increase in the AUC metric, compared with that of the other models. Owing to the limited information in incomplete address descriptions, all models showed a decrease in performance for incomplete address descriptions compared with complete address descriptions. However, the HGAM exhibited markedly lesser performance degradation for

incomplete address descriptions, significantly outperforming the baseline model. Specifically, the mean distance error of the HGAM increased by only 24.28 m, whereas the average distance errors of the baseline models SLG, MLG, and MLSG increased by 51.46 m, 39.31 m, and 44.54 m, respectively. In addition, the median distance error of HGAM increased by only 1.88 m compared to the full address description. This indicates that the proposed model is more robust. By considering both local and global information, it can better cope with missing elements and maintain a relatively good performance even in cases of insufficient information.

Table 6. Performance of SLG, MLG, MLSG, and our model across various address types.

Model	Accuracy@50m				Median				Mean			
	SLG	MLG	MLSG	HGAM	SLG	MLG	MLSG	HGAM	SLG	MLG	MLSG	HGAM
Complete	0.51	0.53	0.51	0.57	48.03	44.93	47.16	39.87	117.52	112.96	93.99	87.11
Incomplete	0.49	0.51	0.48	0.56	51.93	48.05	50.37	41.76	167.98	152.27	138.53	111.39

4.3.5. Ablation Study

As shown in Table 7, incorporating the cross-attention (CA) mechanism resulted in a 3-percentage-point improvement in both the AUC and Accuracy@50m metrics of our model and reductions of 3.87 m and 3.42 m in the median and mean distance errors, respectively. Introducing the layer normalization mechanism led to a 2-percentage-point improvement in both the AUC and Accuracy@50m metrics and reductions of 1.44 m and 2.16 m to the median and mean distance errors, respectively. Additionally, compared to the self-attention (SA) module, using the cross-attention (CA) module improved the AUC and Accuracy@50m metrics for our model by 1 and 2 percentage points, respectively, and reduced the median and mean distance errors by 2.73 m and 3.01 m, respectively. These results demonstrate the effectiveness and superiority of the cross-attention and layer normalization modules.

Table 7. Results of the ablation study.

Model	AUC	Accuracy@50m	Median	Mean
HAGM-CA	0.59	0.56	41.46	93.98
-w/o CA	0.56	0.53	45.33	97.40
-w/o LN	0.57	0.54	42.90	96.14
HAGM-SA	0.58	0.54	44.19	96.99

The cross-attention mechanism enables the model to dynamically perceive and focus on different elements of the address text, thereby accurately capturing the contextual information of the corresponding geographical scale. This enhances the prediction precision by establishing a correspondence between the input address text elements and geographic grids across different scales. The model focuses on large-scale textual description features (e.g., provinces) when predicting characters at the front of the sequence and focuses on finer-scale textual geographic information features (e.g., specific buildings) when predicting characters at the back of the sequence, which matches the hierarchical structure of the address text.

The LN module not only accelerates the training process of the model through the layer normalization process but also provides stability and avoids gradient explosion or vanishing of the model during the training process, which further improves the robustness and prediction accuracy of the model.

4.4. Discussion

In summary, SLG exhibited the lowest performance metrics in all aspects; MLSG stood out in the mean distance error but was relatively weaker in AUC, median distance error, and Accuracy@50m; and MLG performed well in Accuracy@50m and median distance error but fell short in mean distance error. By contrast, our model performed best in all four evaluation metrics—AUC, Accuracy@50m, mean distance error, and median distance error—demonstrating a well-balanced performance.

Compared with the simple single-level geocoding model (SLG), both MLG and MLSG demonstrate partial performance improvements. MLG integrates the classification results of multi-level grids, which, to some degree, mitigates dimensionality explosion and shows enhanced performance in sparse regions. MLSG combines classification and coordinate regression losses and derives the final prediction coordinates by weighting the classification results, which enhances its robustness. However, they still have limitations because their accuracies decrease at finer spatial partitioning scales.

In contrast, our model achieved the best performance for all four metrics, demonstrating a more comprehensive performance. Moreover, it is worth noting that our model has significant advantages in sparse and regular regions, as shown in Table 5 and Figure 4. Taking the evaluation results at level 20 as an example, in sparse and regular regions, HGAM reduced the metric of median distance error by at least about 52 m and 44 m, respectively, compared to other models. Moreover, it showed an increase of at least 23 percentage points in the Accuracy@50m metric compared to the other models. It adopts a sequence-to-sequence approach to predict S2 tokens character by character, which leans towards learning the mapping patterns from specific elements of the text to the corresponding S2 token characters. By integrating the cross-attention and residual connection module, it learns the hierarchical structure of the address text and the output S2 token sequence in more detail in order to establish the correspondence between address text elements at different hierarchical levels and geospatial grids of varying scales. This ensures that the prediction of different locations of S2 tokens dynamically focuses on the corresponding geographical scale information rather than simply predicting the entire grid label category based on the address text as a whole.

Although our model shows good performance, these deep learning methods, including ours, often require new pretraining or fine-tuning in unknown regions, which may lead to increased geocoding costs. In the future, we will expand our study area and explore efficient fine-tuning methods to reduce the construction costs of geocoding models.

5. Conclusions and Future Work

In this study, we modeled geocoding as a sequence-to-sequence task and introduced a grid-based hierarchy-aware geocoding model (HGAM) that incorporated a cross-attention mechanism within the Seq2Seq framework. It predicts the S2 tokens corresponding to the input address text character-by-character and takes the coordinates of the center of the corresponding grid cell as the final predicted geographical location. HAGM aims to "translate" textual language (specific, human-readable address descriptions) into geographical language (precise, machine-recognizable S2 token sequences). Comparative experiments with several mainstream models demonstrated the superior performance of our model for most evaluation metrics, highlighting its accuracy and stability. In particular, our model exhibits better problem-solving capabilities than other models in solving the two major challenges of geocoding—dimensional explosion and inadequate learning from sparse regions—demonstrating the great potential of sequence-to-sequence models in such application scenarios. Furthermore, given that the Seq2Seq model allows for the processing of variable-length input and output sequences, we plan to further explore the dynamic generation of grid labels of corresponding scales based on the address text of different hierarchical levels in the future.

Author Contributions: Conceptualization, Linlin Liang, Yizhuo Quan, and Chengbo Wang; methodology, Linlin Liang; validation, Linlin Liang and Yizhuo Quan; formal analysis, Linlin Liang and Yuanfei Chang; investigation, Linlin Liang and Yizhuo Quan; resources, Linlin Liang, Yuanfei Chang, and Chengbo Wang; data curation, Linlin Liang and Yizhuo Quan; writing—original draft preparation, Linlin Liang; writing—review and editing, Linlin Liang and Chengbo Wang; visualization, Linlin Liang; supervision, Yuanfei Chang; project administration, Yuanfei Chang and Chengbo Wang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2022YFC3301603.

Data Availability Statement: The data presented in this study are available upon request from the corresponding authors. The data are not publicly available due to privacy.

Acknowledgments: We are grateful for the comments and contributions of the anonymous reviewers and the members of the editorial team.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kulkarni, S.; Jain, S.; Hosseini, M.J.; Baldridge, J.; Ie, E.; Zhang, L. Spatial Language Representation with Multi-Level Geocoding. *arXiv preprint* **2020**, arXiv:2008.09236.
2. Viegas, D.A.A. *Toponym Resolution in Text with Neural Language Models*. Master's Thesis, Instituto Superior Técnico, Lisbon, Portugal, 2021.
3. Gritta, M.; Pilehvar, M.T.; Limsopatham, N.; Collier, N. What's Missing in Geographical Parsing? *Lang. Resour. Eval.* **2018**, *52*, 603–623. https://doi.org/10.1007/s10579-017-9385-8.
4. Zhu, X.X.; Wang, Y.; Kochupillai, M.; Werner, M.; Häberle, M.; Hoffmann, E.J.; Taubenböck, H.; Tuia, D.; Levering, A.; Jacobs, N.; et al. Geoinformation Harvesting From Social Media Data: A Community Remote Sensing Approach. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 150–180. https://doi.org/10.1109/MGRS.2022.3219584.
5. Goldberg, D.W.; Wilson, J.P.; Knoblock, C.A. From Text to Geographic Coordinates: The Current State of Geocoding. *Urisa J.* **2007**, *19*, 33–46.
6. Gritta, M.; Pilehvar, M.T.; Collier, N. Which Melbourne? Augmenting Geocoding with Maps. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1285–1296.
7. Zhang, W.; Gelernter, J. Geocoding Location Expressions in Twitter Messages: A Preference Learning Method. *JOSIS* **2014**, *9*, 37–70. https://doi.org/10.5311/JOSIS.2014.9.170.
8. Santos, J.; Anastácio, I.; Martins, B. Using Machine Learning Methods for Disambiguating Place References in Textual Documents. *GeoJournal* **2015**, *80*, 375–392. https://doi.org/10.1007/s10708-014-9553-y.
9. Karimzadeh, M.; Pezanowski, S.; MacEachren, A.M.; Wallgrün, J.O. GeoTxt: A Scalable Geoparsing System for Unstructured Text Geolocation. *Trans. GIS* **2019**, *23*, 118–136. https://doi.org/10.1111/tgis.12510.
10. Lin, Y.; Kang, M.; Wu, Y.; Du, Q.; Liu, T. A Deep Learning Architecture for Semantic Address Matching. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 559–576. https://doi.org/10.1080/13658816.2019.1681431.
11. Hosseini, K.; Nanni, F.; Coll Ardanuy, M. DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 6–20 November 2020; Association for Computational Linguistics: Online, 2020; pp. 62–69.
12. Yao, X. Georeferencing, Geocoding. In *International Encyclopedia of Human Geography*; Kitchin, R., Thrift, N., Eds.; Elsevier: Oxford, UK, 2009; pp. 458–465, ISBN 978-0-08-044910-4.
13. Fornaciari, T.; Hovy, D. Geolocation with Attention-Based Multitask Learning Models. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), Hong Kong, China, 4 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 217–223.
14. Cardoso, A.B.; Martins, B.; Estima, J. *Using Recurrent Neural Networks for Toponym Resolution in Text*; Moura Oliveira, P., Novais, P., Reis, L.P., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11805, pp. 769–780.
15. Huang, J.; Wang, H.; Sun, Y.; Shi, Y.; Huang, Z.; Zhuo, A.; Feng, S. ERNIE-GeoL: A Geography-and-Language Pre-Trained Model and Its Applications in Baidu Maps. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022. https://doi.org/10.48550/arXiv.2203.09127.
16. Serdyukov, P.; Murdock, V.; Van Zwol, R. Placing Flickr Photos on a Map. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 19 July 2009; ACM: Boston, MA, USA, 2009; pp. 484–491.
17. Wing, B.; Baldridge, J. Hierarchical Discriminative Classification for Text-Based Geolocation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 336–348.

18. DeLozier, G.; Baldridge, J.; London, L. Gazetteer-Independent Toponym Resolution Using Geographic Word Profiles. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29. https://doi.org/10.1609/aaai.v29i1.9531.
19. Müller-Budack, E.; Pustu-Iren, K.; Ewerth, R. *Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 11216, pp. 575–592.
20. Weyand, T.; Kostrikov, I.; Philbin, J. PlaNet—Photo Geolocation with Convolutional Neural Networks. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9912, pp. 37–55, ISBN 978-3-319-46483-1.
21. Seo, P.H.; Weyand, T.; Sim, J.; Han, B. *CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Singapore, 2018; Volume 11214, pp. 544–560.
22. Wang, X.; Wang, R.; Zhan, W.; Yang, B.; Li, L.; Chen, F.; Meng, L. A Storage Method for Remote Sensing Images Based on Google S2. *IEEE Access.* **2020**, *8*, 74943–74956. https://doi.org/10.1109/ACCESS.2020.2988631.
23. Fuli, C. A *Full-Text Retrieval Method for Spatial Data Search Based on Global Subdivision Grid*; Geomatics World: Marco Island, FL, USA, 2015.
24. Qiao-hu, D. A Method of Spatial Association for Multi-Sources Remote Sensing Data Based on Global Subdivision Grid. *Sci. Surv. Mapp.* **2015**, *40*, 4.
25. Cepeda, V.V.; Nayak, G.K.; Shah, M. GeoCLIP: Clip-Inspired Alignment between Locations and Images for Effective Worldwide Geo-Localization. *arXiv preprint* **2023**, arXiv:2309.16020. https://doi.org/10.48550/ARXIV.2309.16020.
26. Haas, L.; Skreta, M.; Alberti, S. PIGEON: Predicting Image Geolocations. *arXiv preprint* **2023**, arXiv:2307.05845. https://doi.org/10.48550/ARXIV.2307.05845.
27. Ding, R.; Chen, B.; Xie, P.; Huang, F.; Li, X.; Zhang, Q.; Xu, Y. A Multi-Modal Geographic Pre-Training Method. *arXiv preprint* **2023**, arXiv:2301.04283. https://doi.org/10.48550/ARXIV.2301.04283.
28. Anonymous. A Survey on Geocoding: Algorithms and Datasets for Toponym Resolution. *ACL ARR 2021 November Blind Submission, 17 Nov 2021 (modified: 06 May 2023).* https://openreview.net/pdf?id=-koTfmSDsM.
29. Kulkarni, S.; Jain, S.; Hosseini, M.J.; Baldridge, J.; Ie, E.; Zhang, L. Multi-Level Gazetteer-Free Geocoding. In Proceedings of the Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics, Online, 5-6 August 2021; pp. 79–88. https://doi.org/10.18653/v1/2021.splurobonlp-1.9.
30. Li, F.; Lu, Y.; Mao, X.; Duan, J.; Liu, X. Multi-Task Deep Learning Model Based on Hierarchical Relations of Address Elements for Semantic Address Matching. *Neural Comput. Applic.* **2022**, *34*, 8919–8931. https://doi.org/10.1007/s00521-022-06914-1.
31. Leidner, J.L. Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding. In *SIGIR Forum*; ACM: New York, NY, USA, 2007; Volume 41, pp. 124–126. https://doi.org/10.1145/1328964.1328989.
32. Karimzadeh, M.; Huang, W.; Banerjee, S.; Wallgrün, J.O.; Hardisty, F.; Pezanowski, S.; Mitra, P.; MacEachren, A.M. GeoTxt: A Web API to Leverage Place References in Text. In Proceedings of the 7th Workshop on Geographic Information Retrieval, Orlando, FL, USA, 5 November 2013; ACM: Orlando, FL, USA, 2013; pp. 72–73.
33. Lieberman, M.D.; Samet, H. Adaptive Context Features for Toponym Resolution in Streaming News. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, OR, USA, 12 August 2012; ACM: Portland, OR, USA, 2012; pp. 731–740.
34. Radford, B.J. Regressing Location on Text for Probabilistic Geocoding. *arXiv preprint* **2021**, arXiv:2107.00080.
35. Liu, J.; Inkpen, D. Estimating User Location in Social Media with Stacked Denoising Auto-Encoders. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, Colorado, 5 June 2015; Blunsom, P., Cohen, S., Dhillon, P., Liang, P., Eds.; Association for Computational Linguistics: Denver, Colorado, 2015; pp. 201–210.
36. Fize, J.; Moncla, L.; Martins, B. Deep Learning for Toponym Resolution: Geocoding Based on Pairs of Toponyms. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 818. https://doi.org/10.3390/ijgi10120818.
37. Xu, L.; Du, Z.; Mao, R.; Zhang, F.; Liu, R. GSAM: A Deep Neural Network Model for Extracting Computational Representations of Chinese Addresses Fused with Geospatial Feature. *Comput. Environ. Urban. Syst.* **2020**, *81*, 101473. https://doi.org/10.1016/j.compenvurbsys.2020.101473.
38. Laparra, E.; Bethard, S. A Dataset and Evaluation Framework for Complex Geographical Description Parsing. In Proceedings of the 28th International Conference on Computational Linguistics; International Committee on Computational Linguistics: Barcelona, Spain (Online), 8–13 December 2020; pp. 936–948.
39. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
40. Cho, K.; Van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1724–1734.
41. Rush, A.M.; Chopra, S.; Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 379–389.

42. Hermann, K.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. *arXiv* **2015**, arXiv:1506.03340.
43. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964. https://doi.org/10.1109/ICASSP.2016.7472621.
44. Vinyals, O.; Le, Q.V. A Neural Conversational Model. *arXiv* **2015**, arXiv:1506.05869.
45. Yin, P.; Neubig, G. A Syntactic Neural Model for General-Purpose Code Generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 440–450.
46. Zhang, Z.; Li, M.; Lin, X.; Wang, Y.; He, F. Multistep Speed Prediction on Traffic Networks: A Graph Convolutional Sequence-to-Sequence Learning Approach with Attention Mechanism. *arXiv* **2018**, arXiv:1810.10237.
47. Zhong, V.; Xiong, C.; Socher, R. Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning. *arXiv* **2017**, arXiv:1709.00103.
48. Qian, C.; Yi, C.; Cheng, C.; Pu, G.; Liu, J. A Coarse-to-Fine Model for Geolocating Chinese Addresses. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 698. https://doi.org/10.3390/ijgi9120698.
49. Cheng-Qi, C. Spatial Data Coding Method Based on Global Subdivision Grid. *J. Geomat. Sci. Technol.* **2013**, *30*, 284–287.
50. Ekawati, R.; Suprihadi, U. Analysis of S2 (Spherical) Geometry Library Algorithm for GIS Geocoding Engineering. *TELKOMNIKA* **2018**, *16*, 334. https://doi.org/10.12928/telkomnika.v15i4.6985.
51. Kamalloo, E.; Rafiei, D. A Coherent Unsupervised Model for Toponym Resolution. In Proceedings of the 2018 World Wide Web Conference on World Wide Web—WWW'18, Lyon, France, 23–27 April 2018; ACM Press: Lyon, France, 2018; pp. 1287–1296.
52. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 657–668.
53. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint* **2019**, arXiv:1907.11692.
54. Ba, J.; Kiros, J.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
55. Jurgens, D.; Finethy, T.; McCorriston, J.; Xu, Y.; Ruths, D. Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice. In Proceedings of the International AAAI Conference on Web and Social Media, Oxford, UK, 26–29 May 2015; Volume 9, pp. 188–197. https://doi.org/10.1609/icwsm.v9i1.14627.
56. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv preprint* **2017**, arXiv:1412.6980v9.