

OSMT: Bridging OpenStreetMap Queries and Natural Language with Open-source Tag-aware Language Models

Zhuoyue Wan*, Wentao Hu*, Chen Jason Zhang*, Yuanfeng Song^{†¶}, Shuaimin Li[‡],
Ruiqiang Xiao[§], Xiao-Yong Wei*, Raymond Chi-Wing Wong[§]

*The Hong Kong Polytechnic University, Hong Kong, China [†]WeBank, Shenzhen, China

[‡]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[§]The Hong Kong University of Science and Technology, Hong Kong, China

Abstract—Bridging natural language and structured query languages is a long-standing challenge in the database community. While recent advances in language models have shown promise in this direction, existing solutions often rely on large-scale closed-source models that suffer from high inference costs, limited transparency, and lack of adaptability for lightweight deployment. In this paper, we present OSMT, an open-source tag-aware language model specifically designed to bridge natural language and Overpass Query Language (OverpassQL), a structured query language for accessing large-scale OpenStreetMap (OSM) data. To enhance the accuracy and structural validity of generated queries, we introduce a Tag Retrieval Augmentation (TRA) mechanism that incorporates contextually relevant tag knowledge into the generation process. This mechanism is designed to capture the hierarchical and relational dependencies present in the OSM database, addressing the topological complexity inherent in geospatial query formulation. In addition, we define a reverse task, OverpassQL-to-Text, which translates structured queries into natural language explanations to support query interpretation and improve user accessibility. We evaluate OSMT on a public benchmark against strong baselines and observe consistent improvements in both query generation and interpretation. Despite using significantly fewer parameters, our model achieves competitive accuracy, demonstrating the effectiveness of open-source pre-trained language models in bridging natural language and structured query languages within schema-rich geospatial environments.

Index Terms—structured query generation, natural language interfaces, Text-to-OverpassQL, OverpassQL-to-Text, language model

I. INTRODUCTION

Structured query languages are essential interfaces for managing and interacting with complex databases. Establishing effective alignment between natural language and structured queries has emerged as a prominent research focus in both academia and industry, motivated by the need to lower expertise barriers and facilitate intuitive database access for non-technical users. Significant research addressing this challenge has been presented in a broad literature, including works such as [1]–[12]. These studies collectively demonstrate sustained scholarly and practical interest in advancing natural language interfaces for structured data and have been widely adopted

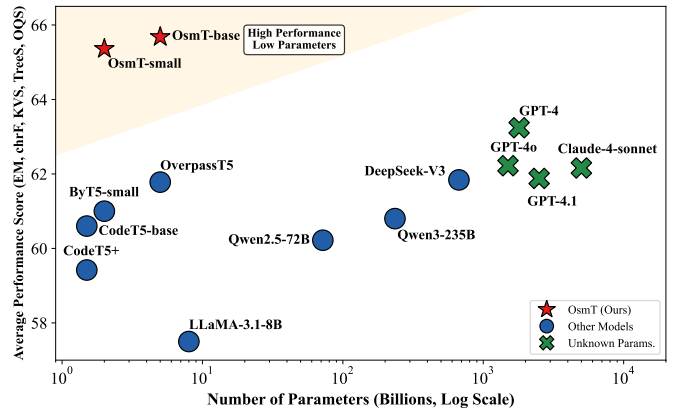


Fig. 1: Model performance vs. parameter size (log scale). Comparison of OSMT with state-of-the-art open-source and closed-source models on the Text-to-OverpassQL task. Average performance is over five metrics.

across diverse real-world applications, ranging from business analytics to scientific data management. Among the various types of structured data, geospatial databases have emerged as particularly critical due to their foundational role in supporting large-scale spatial analysis, complex query execution, and location-based decision-making. These capabilities underpin a wide range of downstream applications, including geospatial knowledge extraction, urban mobility modeling, and spatio-temporal forecasting [13]–[19].

A prominent example of a geospatial database is OpenStreetMap (OSM), a collaboratively maintained, open-access platform that provides foundational infrastructure for spatial data analysis. OSM supports various sophisticated spatial query functionalities, including location filtering, proximity searches, and routing, as exemplified by widely used applications such as OsmAnd¹ and Locus Map². Retrieving structured geospatial data from OSM typically relies on the Overpass Query Language (OverpassQL), a domain-specific language designed for fine-grained spatial data extraction through filters, scoped conditions, and recursive constructs.

¹<https://osmand.net/>

²<https://web.locusmap.app/en/>

[¶] Corresponding author.

While OverpassQL offers powerful expressive capabilities, it demands that users possess detailed knowledge of OSM’s schema and tagging structure, creating significant usability barriers for non-expert users.

To improve the accessibility of OSM data, the Text-to-OverpassQL task [20] has been proposed to translate natural language queries into executable OverpassQL statements, enabling more intuitive access to structured spatial data. However, existing approaches still face several key limitations. Many recent methods are based on proprietary large-scale language models that require significant computational resources, which restricts their deployment in real-world geospatial systems with limited capacity. In addition, current models typically treat query generation as a generic sequence-to-sequence problem, without explicitly modeling the hierarchical structure and relational semantics encoded in the OSM tag schema. Moreover, most systems lack support for the reverse task of translating OverpassQL statements into coherent and informative natural language, which is essential for improving the interpretability and usability of structured query interfaces.

To address the aforementioned challenges, we present **OSMT**, an open-source tag-aware pre-trained language model designed for translating between natural language and structured OverpassQL queries. As shown in Figure 1, while existing models vary widely in parameter size and performance, our approach achieves consistently strong results under a comparable or significantly smaller parameter budget. These results highlight the effectiveness of OSMT in delivering both accuracy and efficiency, making it well suited for practical deployment in geospatial applications.

Previous studies typically formulate Text-to-OverpassQL as a generic sequence-to-sequence (Seq2Seq) generation problem, directly mapping natural language inputs to structured queries without explicitly leveraging structural information embedded in the OSM tagging schema, as illustrated in Figure 2. In contrast, our approach explicitly captures the hierarchical and relational dependencies inherently encoded within OSM to enhance spatial reasoning capabilities. Specifically, we propose a Tag Retrieval Augmentation (TRA) mechanism that incorporates relevant tag knowledge derived from OSM’s structured organization of spatial entities, including nodes, ways, and relations. Given a natural language query such as “All mountain peaks, boundary stones, and alpine huts in the current view.” (Figure 2), our TRA mechanism effectively identifies implicit semantic associations between query terms and corresponding OSM tags, for example, associating “mountain peaks” with the tag *natural*, “boundary stones” with *historic*, and “alpine huts” with *tourism*. These associations, although implicit in user queries, are explicitly defined by the OSM schema. By utilizing this structured tag knowledge, our method achieves improved accuracy and generates context-aware OverpassQL queries that better align with user intent.

Given the steep learning curve associated with OverpassQL, automating the translation from natural language to query statements addresses only part of the challenge. A robust

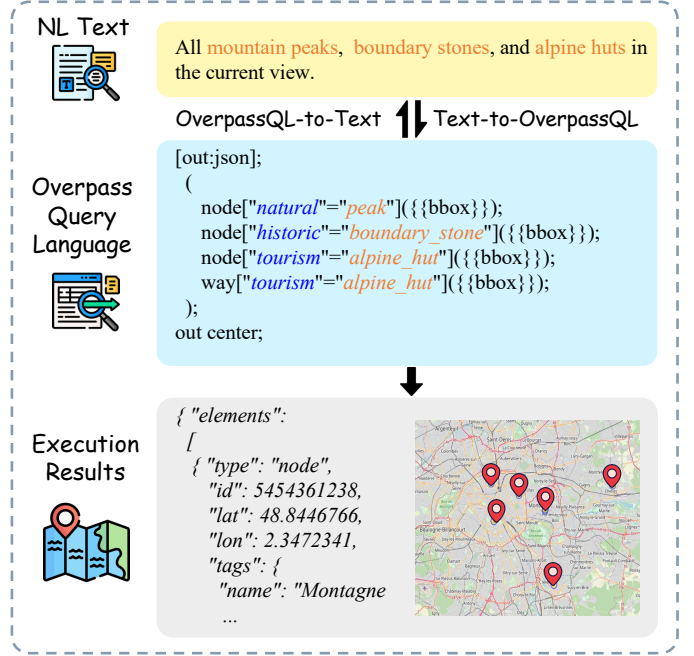


Fig. 2: Overview of the Text-to-OverpassQL and OverpassQL-to-Text tasks. Illustration of the bidirectional translation between natural language and OverpassQL, along with the corresponding query execution output and the visualization of retrieved entities on the map.

model must not only understand user intent but also produce syntactically valid queries that conform to OverpassQL’s strict grammar, as illustrated in Figure 2. To support this, we construct a task-specific pre-training corpus that combines structured OverpassQL queries with natural language inputs and OSM tag information. This joint representation enables the model to acquire both semantic alignment and syntactic fluency during pre-training. In addition to forward generation, we introduce a complementary reverse task, OverpassQL-to-Text, which translates structured queries into coherent and informative natural language descriptions. This bidirectional capability enhances the interpretability, accessibility, and overall usability of structured geospatial query systems.

Beyond accuracy and interpretability, practical deployment requires models that are transparent, efficient, and suitable for lightweight integration, especially in large-scale, dynamically evolving spatial databases. Unlike closed-source language models, which often entail prohibitive inference costs and limited adaptability, our open-source model design is tailored for geospatial query tasks and prioritizes computational efficiency and transparency. Through targeted pre-training on task-relevant corpora that combine natural language texts, OSM tagging semantics, and structured OverpassQL, OSMT achieves competitive performance while maintaining a substantially smaller parameter footprint. This combination of open-source transparency, efficiency, and scalability positions OSMT as particularly suitable for real-world geospatial applications, including resource-constrained environments.

In summary, our main contributions are:

- We propose **OSMT**, the first open-source pre-trained language model specifically designed for bridging natural language queries and structured OverpassQL statements. OSMT integrates natural language text, structured OverpassQL syntax, and OSM tag semantics through a hybrid pre-training strategy tailored explicitly for geospatial querying tasks.
- We introduce a novel **Tag Retrieval Augmentation (TRA)** mechanism that explicitly leverages hierarchical and relational tag knowledge embedded within the OSM database schema. By incorporating this structured context into query generation, TRA substantially enhances the semantic accuracy and structural coherence of Text-to-OverpassQL translations.
- We define a complementary reverse task, **OverpassQL-to-Text**, to translate structured queries into clear and informative natural language descriptions. This bidirectional design significantly improves query interpretability and accessibility, benefiting non-expert users interacting with complex spatial databases.
- We conducted extensive experiments on publicly available benchmarks for both Text-to-OverpassQL and OverpassQL-to-Text tasks. Experimental results demonstrate that OSMT consistently surpasses strong baseline models in terms of accuracy, interpretability, and parameter efficiency, setting new standards for natural language interfaces within geospatial database systems.

II. PRELIMINARIES

To facilitate seamless interaction between natural language and structured geospatial queries, we define two complementary tasks: the established *Text-to-OverpassQL* task and its inverse, *OverpassQL-to-Text*.

A. Text-to-OverpassQL

This task aims to translate a natural language query q into a syntactically correct and semantically meaningful OverpassQL statement Ovq . Unlike prior work [20], which formulates this as a standard Seq2Seq translation problem, we explicitly encode the topological structures inherent in the OSM database to enhance spatial reasoning. Formally, let $\mathcal{D}$ denote the OSM database containing structured spatial entities. The generation process is defined as:

$$Ovq = f(q, \mathcal{D}),$$

where f is the underlying generation model.

B. OverpassQL-to-Text

This task performs the reverse mapping by taking a structured OverpassQL script Ovq , typically authored by technical users, and generating a coherent natural language explanation e . The generation process is given by:

$$e = f(Ovq).$$

In both tasks, the generation model f is implemented using our open-source language model LM_θ , which is specifically

pre-trained and fine-tuned to capture the semantic alignment between natural language and OverpassQL. The model supports bidirectional mappings as follows:

$$Ovq = LM_\theta(q, \mathcal{D}) \quad \text{and} \quad e = LM_\theta(Ovq),$$

where θ represents the shared model parameters.

III. METHODOLOGY

Osmt Framework Overview: Figure 3 presents an overview of the OSMT framework, comprising four core components: pre-training corpus collection and construction (Section III-A), hybrid objective pre-training (Section III-B), supervised fine-tuning (Section III-C), and tag retrieval augmentation (Section III-D).

A. Pre-training Corpus Collection and Construction

To support cross-modal pre-training that bridges natural language, symbolic tag knowledge, and structured OverpassQL, we construct a pre-training corpus from three complementary sources. Each contributes a distinct supervision signal to promote structural awareness, semantic grounding, and bidirectional generation capabilities.

OSM Taginfo. To encode symbolic representations of geospatial semantics, we extract (*key*, *relation*, *value*) tag triples from OSM Taginfo³, a community-curated platform that aggregates global statistics on tag usage, frequency, and co-occurrence. This resource provides a comprehensive and realistic tag vocabulary reflecting actual editing behavior in OSM. It serves as the lexical foundation for symbolic query understanding and generation.

Tag-Descriptions. To bridge symbolic tags with natural language semantics, we collect 11,880 (*tag*, *description*) pairs from the OSM Wiki. These descriptions provide human-readable explanations of tag meanings and typical usage contexts. By aligning tags with textual descriptions during pre-training, the model learns to associate symbolic inputs with their semantic interpretations, enhancing both interpretability and contextual understanding in downstream generation.

OverpassNL. The OverpassNL dataset [20] serves as a parallel supervision source for both directions of structured query modeling. It contains 8,352 natural language questions paired with real-world OverpassQL queries, spanning diverse geographic intents and varying levels of query complexity. Each pair provides a bidirectional alignment between user intent and OverpassQL syntax, supporting both Text-to-OverpassQL generation and OverpassQL-to-Text interpretation. Incorporating this dataset into the pre-training corpus enables the model to learn consistent mappings between natural language and OverpassQL structures.

As illustrated in Figure 3, we construct the pre-training corpus through a systematic pipeline that integrates the three data sources and partitions them into six data types: OverpassQL queries (20%), natural language questions (20%), individual tags (25.4%), tag descriptions (7.3%), Tag-Description

³<https://taginfo.openstreetmap.org/>

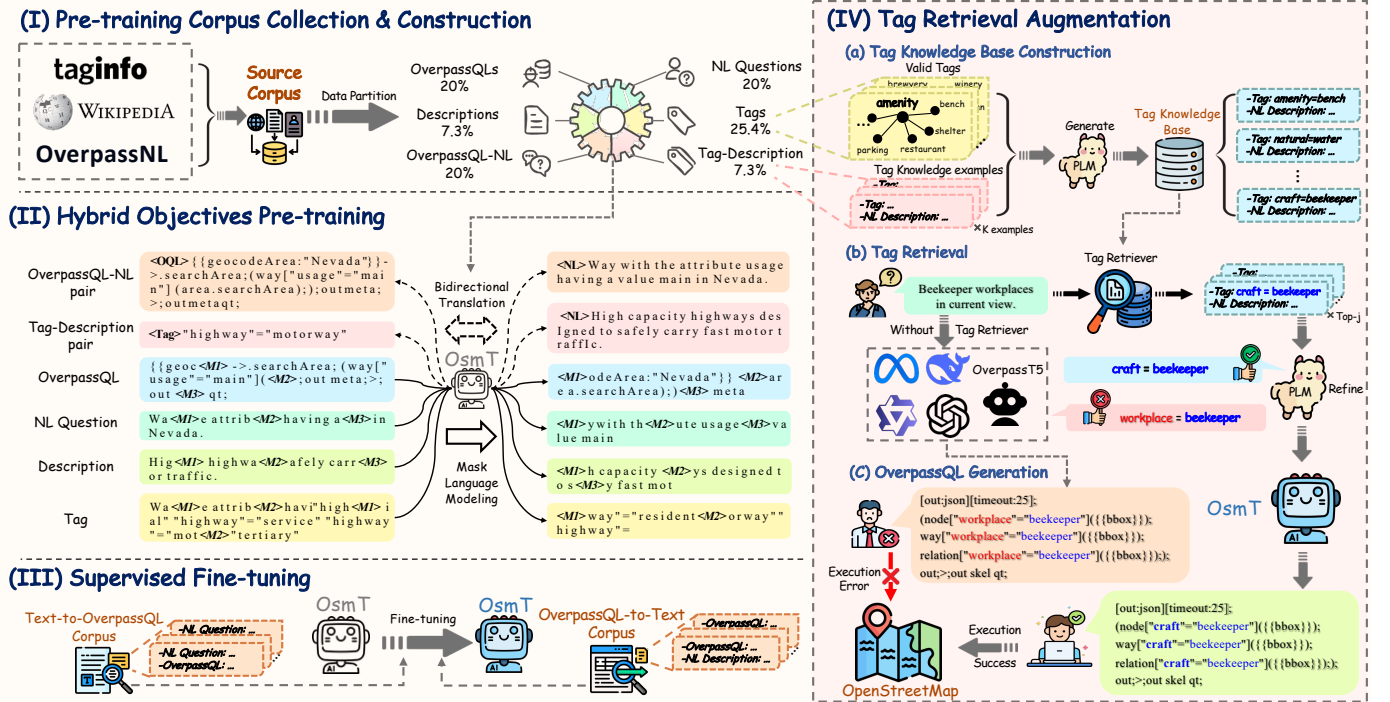


Fig. 3: Overview of the OSMT framework, which includes (I) pre-training corpus collection and construction, (II) hybrid pre-training with MLM and BT objectives, (III) supervised fine-tuning, and (IV) tag retrieval augmentation.

pairs (7.3%), and OverpassQL–NL pairs (20%). To ensure structural consistency across these heterogeneous inputs, we normalize all (key, relation, value) tag triples into a unified format compatible with OverpassQL syntax, namely ('key', relation, 'value'). This normalization bridges the gap between statistical tag usage patterns and the syntactic constraints of OverpassQL, and is applied to tags collected from both the OSM Taginfo and tag description resources. We further apply corpus-level preprocessing, including deduplication and syntax correction, to enhance data quality and consistency. To prevent data leakage during downstream fine-tuning, we adopt the original OverpassNL split strategy to separate training and validation sets.

B. Hybrid Objectives Pre-training

Masked Language Modeling. We adopt the span corruption variant of the Masked Language Modeling (MLM) objective for pretraining our open-source language model. Specifically, we follow the byte-level strategy from ByT5 [21], where continuous spans of subword tokens are randomly masked and replaced with sentinel tokens (e.g., $\langle M1 \rangle$, $\langle M2 \rangle$) as shown in Figure 3. The decoder is then tasked with reconstructing the masked spans in order, each preceded by the corresponding sentinel token. This objective encourages the model to capture both local and long-range dependencies in cross-modal inputs. Formally, for an input sequence $\mathbf{x}$ of N tokens with masked tokens $\mathbf{x}^m$ and visible tokens $\mathbf{x}^{\setminus m}$, the MLM loss is defined as:

$$\mathcal{L}_{\text{MLM}}(\theta) = \sum_{n=1}^N -\log P_{\theta} \left(x_n^m \mid \mathbf{x}^{\setminus m}, \mathbf{x}_{<n}^m \right), \quad (1)$$

where θ denotes the model parameters, x_n^m is the n -th masked token to be predicted, and $\mathbf{x}_{<n}^m$ represents the sequence of previously generated masked tokens.

Bidirectional Translation. To better align pre-training with downstream generation tasks, we introduce a Bidirectional Translation (BT) objective over cross-modal corpora. Unlike conventional unidirectional translation, our approach treats each paired sequence of OverpassQL and natural language, or tag and description, as mutually translatable in both directions. Specifically, for every sequence pair (s, t) , we construct training examples in both source-to-target and target-to-source directions. Given a target sequence $\mathbf{t} = \{t_1, t_2, \dots, t_T\}$, the BT objective is defined as:

$$\mathcal{L}_{\text{BT}}(\theta) = \sum_{i=1}^T -\log P_{\theta}(t_i \mid \mathbf{s}, \mathbf{t}_{<i}), \quad (2)$$

where $\mathbf{s}$ denotes the source sequence (either a natural language query, OverpassQL query, or tag/description), and $\mathbf{t}_{<i}$ represents the prefix tokens generated up to step i . This bidirectional formulation enhances the model's ability to learn semantically grounded mappings between natural language and structured representations.

Hybrid Objectives. To combine the strengths of masked language modeling and bidirectional translation, we construct a hybrid pre-training objective by interleaving examples from both tasks. During training, each mini-batch is sampled from

a heterogeneous cross-modal corpus that includes natural language inputs, OverpassQL queries, and OSM tag data. The overall training objective is defined as the sum of the two component losses:

$$\mathcal{L}_{\text{Hybrid}}(\theta) = \mathcal{L}_{\text{MLM}}(\theta) + \mathcal{L}_{\text{BT}}(\theta). \quad (3)$$

Figure 3 illustrates the Hybrid Pre-training Objectives. The solid-line segments denote the MLM objective, applied across multiple modalities including OverpassQL queries, natural language questions, individual tags, and tag descriptions. Masked spans are replaced by sentinel tokens and reconstructed by the decoder. The dashed-line segments represent the Bidirectional Translation objective, encompassing dual-directional generation between OverpassQL and natural language, as well as between tags and descriptions. This hybrid design enables OSMT to capture structural fidelity, semantic alignment, and contextual reasoning across heterogeneous input sources.

C. Supervised Fine-Tuning

Supervised fine-tuning (SFT) further adapts the pretrained OSMT model to the two downstream tasks of Text-to-OverpassQL and OverpassQL-to-Text. We follow the original train, development, and test splits defined in prior work [20] to ensure consistent evaluation. During fine-tuning, the model is trained to generate a target sequence, which may be either a structured OverpassQL query or a natural language explanation, conditioned on a corresponding input sequence, which may be a natural language question or an OverpassQL statement. Given a source sequence $\mathbf{s}$ and a target sequence $t_{i=1}^T$, the objective minimizes the negative log-likelihood of the target tokens:

$$\mathcal{L}_{\text{SFT}}(\theta) = \sum_{i=1}^T -\log P_{\theta}(t_i | \mathbf{s}, \mathbf{t}_{<i}), \quad (4)$$

where θ denotes the model parameters, $\mathbf{s}$ is the input sequence, and $\mathbf{t}_{<i}$ represents the prefix of the target sequence generated up to step i . This objective follows the same Seq2Seq paradigm used in pre-training, ensuring consistency across both learning stages. The loss is averaged over all training examples during optimization.

D. Tag Retrieval Augmentation

Tags in OverpassQL are essential for specifying the attributes of spatial entities, allowing users to filter and retrieve map features based on key-value constraints. However, for users without prior domain expertise, identifying the appropriate tag keys and values from natural language input is often non-trivial. The ambiguity and incompleteness of natural language expressions can result in incorrect or invalid tag selections, thereby compromising query correctness.

As illustrated in Figure 3, existing models such as OverpassT5 [20] and even advanced Large Language Models (LLMs), often fail to accurately interpret tag semantics in realistic scenarios. For example, given the input query ‘‘Beekeeper workplaces in the current view,’’ these models mistakenly

Algorithm 1 Tag Knowledge Base Construction

Require: Dataset $\mathcal{D}$ containing query-tag pairs $(\mathbf{q}, \mathbf{t})$; Dataset $\mathcal{V}$ containing all valid tags $\mathbf{t}$; Pre-trained Language Model PLM ; Prompt templates $\mathcal{T}$; Embedding function g ;

Ensure: Tag knowledge base $\mathcal{K}$

```

1:  $\mathcal{K} \leftarrow \{\}$   $\triangleright$  Initialize an empty tag knowledge base
2: for all  $\mathbf{k} = (\mathbf{q}, \mathbf{t}) \in \mathcal{D}$  do
3:    $\mathcal{K} \leftarrow \mathcal{K} \cup \mathbf{k}$   $\triangleright$  Add existing tag knowledge
4: end for
5: for all  $\mathbf{t} \in \mathcal{V}$  do
6:    $\mathcal{E} \leftarrow$  Fixed  $k$ -examples tag knowledge from  $\mathcal{D}$ 
7:    $\mathbf{q}_{\text{gen}} \leftarrow \text{PLM}(\mathcal{T}_{\text{gen}}(\mathbf{t}, \mathcal{E}))$   $\triangleright$  Generate corresponding query
8:    $\mathbf{k}_{\text{gen}} \leftarrow (\mathbf{q}_{\text{gen}}, \mathbf{t})$   $\triangleright$  Generate tag knowledge
9:    $\mathcal{K} \leftarrow \mathcal{K} \cup \mathbf{k}_{\text{gen}}$   $\triangleright$  Store tag knowledge
10: end for
11: for all  $\mathbf{k} = (\mathbf{q}, \mathbf{t}) \in \mathcal{K}$  do
12:    $\mathbf{q}_{\text{emb}} \leftarrow g(\mathbf{q})$   $\triangleright$  Compute embedding
13:    $\mathbf{k} \leftarrow (\mathbf{q}, \mathbf{q}_{\text{emb}}, \mathbf{t})$   $\triangleright$  Update tag knowledge
14: end for

```

associate the term ‘‘Beekeeper’’ with the tag key `workplace`, which is not valid in the OSM schema. In contrast, the correct key is `craft`, and the valid tag is `craft=beekeeper`. This failure highlights the limitations of current models in capturing the structured taxonomy and semantic constraints inherent in the OSM tagging system.

To address the limitations arising from incomplete or inaccurate tag specification in user queries, we propose a Tag Retrieval Augmentation (TRA) mechanism that injects symbolic tag knowledge into the OverpassQL generation process. TRA incorporates a semantic embedding-based retriever and a curated tag knowledge base to identify contextually relevant and syntactically valid tags.

Let $\mathbf{t}$ denote the tag set retrieved for a given input query $\mathbf{q}$. This set can be viewed as a filtered and semantically grounded subset of the OSM database $\mathcal{D}$, specialized for query intent understanding. Incorporating $\mathbf{t}$ into the generation process refines the original Text-to-OverpassQL formulation as:

$$\text{Ovq} = \text{PLM}_{\theta}(\mathbf{q}, \mathbf{t}).$$

1) *Tag Knowledge Base Construction:* To support accurate tag retrieval during query generation, we construct a comprehensive tag knowledge base, denoted as $\mathcal{K}$. The construction process is outlined in Algorithm 1. We begin by populating $\mathcal{K}$ with observed query-tag pairs $(\mathbf{q}, \mathbf{t})$ extracted from the OverpassNL training dataset $\mathcal{D}$. Each OverpassQL Ovq is parsed via the Overpass API to obtain its XML representation, from which we extract structured tags $\mathbf{t}$ in the form of key, value, or key-value pairs. To enhance coverage, we supplement $\mathcal{K}$ with generated query-tag pairs for tags in the valid tag set $\mathcal{V}$ that may be underrepresented or unseen in $\mathcal{D}$. For each tag $\mathbf{t} \in \mathcal{V}$, we sample k in-context examples $\mathcal{E} \subset \mathcal{D}$ and apply a pre-trained language model (PLM) to generate a corresponding pseudo-query $\mathbf{q}_{\text{gen}}$ using a prompt constructed via the generation template $\mathcal{T}_{\text{gen}}(\mathbf{t}, \mathcal{E})$.

$\mathcal{T}_{\text{gen}}(t, \mathcal{E})$: Given a target tag t used in OpenStreetMap and a few example (description, tag) pairs $\mathcal{E} = \{(q^i, t^i)_{i=1}^k\}$, generate a natural language description that accurately captures the semantics of the target tag based on the patterns observed in the examples.

The generated pair (q_{gen}, t) is then added to the knowledge base. Finally, to enable semantic matching during retrieval, each query q in $\mathcal{K}$ is encoded into a dense embedding vector $q_{\text{emb}} = g(q)$ using a pre-trained embedding function g . Each knowledge entry is thus stored as a triplet (q, q_{emb}, t) in $\mathcal{K}$. The resulting knowledge base captures both symbolic and semantic representations of tag-query associations, enabling context-aware tag retrieval during inference.

2) *Tag Retrieval*: Given a natural language query q , we first apply the tag retrieval mechanism (Algorithm 2) to retrieve semantically relevant OSM tags from a pre-constructed tag knowledge base $\mathcal{K}$. Specifically, the query q is encoded into a dense embedding q_{emb} using a pretrained embedding model g . We then compute cosine similarities between q_{emb} and all stored embeddings q_{emb}^i in $\mathcal{K}$. The top- j most similar entries are selected, and their associated tag candidates $\{t^i\}_{i=1}^j$ are aggregated to form the initial retrieval set t_{ret} .

3) *Tag Augmentation*: To improve precision and reduce retrieval noise, we employ the same PLM as mentioned before, which is prompted with both the user query q and the retrieved tag set t_{ret} . This produces a refined tag set t_{ref} that is more accurately aligned with the semantic intent of the input. We then filter t_{ref} against the set of valid OSM tags $\mathcal{V}$ to obtain the final tag set t_{valid} , ensuring syntactic correctness.

$\mathcal{T}_{\text{ref}}(q, t_{\text{ret}})$: Given a user query q and a set of potentially relevant OpenStreetMap tags t_{ret} , generate a refined tag set by removing semantically redundant or irrelevant entries and returning a tag subset more consistent with the user intent.

Finally, the validated tag set t_{valid} and the original query q are jointly passed to our proposed model OSM-T, which generates the corresponding OverpassQL query Ovq . By unifying retrieval, refinement, and validation, this tag-aware generation pipeline strengthens structural fidelity and enhances contextual relevance in Text-to-OverpassQL translation.

IV. EXPERIMENTAL SETUP

A. Implementation Details

We adopt ByT5 [21] as the starting point for pretraining due to its open-source availability, compact parameter size, and byte-level tokenization, which ensures compatibility with OverpassQL syntax. We pre-trained OSM-T for 30 epochs on a single NVIDIA H20-NVLink GPU (96GB). For the Text-to-OverpassQL task, we followed the fine-tuning configuration of [20], utilizing four NVIDIA RTX 4090 GPUs (24GB each). In the TRA pipeline, we employ SimCSE [22] for contrastive learning on top of an sBERT embedding model to improve tag

Algorithm 2 Tag Retrieval Augmentation Text-to-OverpassQL

Require: User input query q ; Knowledge base $\mathcal{K}$; Dataset $\mathcal{V}$ containing all valid tags t ; Pre-trained Language model PLM; Prompt templates $\mathcal{T}$; Embedding function g ; Our proposed model OSM-T;
Ensure: OverpassQL statement Ovq

```

1: function RETRIEVER( $q, \mathcal{K}$ )
2:    $q_{\text{emb}} \leftarrow g(q)$  ▷ Compute embedding
3:   for all  $k^i = (q^i, q_{\text{emb}}^i, t^i) \in \mathcal{K}$  do
4:      $Sim^i \leftarrow \text{CosSim}(q_{\text{emb}}, q_{\text{emb}}^i)$  ▷ Calculate similarity
5:   end for
6:    $\{k^i\}_{i=1}^j \leftarrow \text{Top}(\{Sim^i\}_{i=1}^{|\mathcal{K}|}, j)$  ▷ Top- $j$  tag knowledge
7:    $t_{\text{ret}} \leftarrow \{t^i\}_{i=1}^j \in \{k^i\}_{i=1}^j$  ▷ Retrieve corresponding tags
8:    $t_{\text{ref}} \leftarrow \text{PLM}(\mathcal{T}_{\text{ref}}(q, t_{\text{ret}}))$  ▷ Refine tags
9:   return  $t_{\text{ref}}$ 
10: end function
11: function TRA( $q, \mathcal{K}$ )
12:    $t_{\text{ref}} \leftarrow \text{RETRIEVER}(q, \mathcal{K})$ 
13:    $t_{\text{valid}} \leftarrow \mathcal{V} \cap t_{\text{ref}}$  ▷ Filter invalid tags
14:    $Ovq \leftarrow \text{OSMT}(q, t_{\text{valid}})$  ▷ Generate OverpassQL
15:   return  $Ovq$ 
16: end function

```

retrieval, utilizing LLaMA-3.1-8B [23] merely as an auxiliary filter to mitigate retrieval noise. For the OverpassQL-to-Text task, we performed a grid search and selected the optimal settings based on METEOR scores on the validation set. For in-context learning experiments, demonstration retrieval was conducted using the sBERT embedding model [24].

B. Evaluation Metrics

We adopt a comprehensive set of evaluation metrics to assess model performance on both the Text-to-OverpassQL and OverpassQL-to-Text tasks. For the Text-to-OverpassQL task, we follow the evaluation protocol of [20], reporting Exact Match (EM), which requires the predicted OverpassQL query to exactly match the ground truth, and Character F-score (chrF) [25], which captures character-level n-gram overlap and is suitable for syntax-sensitive languages like OverpassQL. To assess semantic correctness, we use Key Value Similarity (KVS), measuring the normalized overlap of key-value pairs, keys, and values between the generated query Ovq_g and the reference query Ovq_r , defined as:

$$\text{KVS}(Ovq_g, Ovq_r) = \frac{|\text{KV}(Ovq_g) \cap \text{KV}(Ovq_r)|}{\max(|\text{KV}(Ovq_g)|, |\text{KV}(Ovq_r)|)}. \quad (5)$$

For structural similarity, we compute Tree Similarity (TreeS) by converting queries into XML trees, stripping variable names and key-value contents, and recursively comparing subtrees. Overpass Query Similarity (OQS) is reported as the average of chrF, KVS, and TreeS, thus capturing surface-level, semantic, and syntactic similarities.

To further evaluate correctness at the execution level, we report Execution Accuracy (EX) and Soft Execution Accuracy (EX_{SOFT}). EX indicates whether the generated query and the reference query return exactly the same set of elements in OpenStreetMap, producing a binary score of 0 or 1. In contrast, EX_{SOFT} measures the degree of overlap between the two result sets, normalized by the maximum number of

TABLE I: Performance comparison on the Text-to-OverpassQL task using the OverpassNL validation set. **TRA** denotes Tag Retrieval Augmentation. The best results are highlighted in **bold**. * means results from [20].

Model	Setting	#Params.	EM $\uparrow$	chrF $\uparrow$	KVS $\uparrow$	TreeS $\uparrow$	OQS $\uparrow$
<i>Finetuning</i>							
CodeT5-base*		220M	19.8 $\pm$ 0.4	74.6 $\pm$ 0.0	63.2 $\pm$ 0.4	73.0 $\pm$ 0.1	70.3 $\pm$ 0.2
CodeT5-base*	+comments	220M	20.3 $\pm$ 0.2	74.9 $\pm$ 0.1	63.6 $\pm$ 0.5	73.5 $\pm$ 0.1	70.7 $\pm$ 0.2
CodeT5+		220M	18.6 $\pm$ 0.7	74.0 $\pm$ 0.3	62.5 $\pm$ 0.7	71.9 $\pm$ 0.3	69.4 $\pm$ 0.4
CodeT5+	+comments	220M	19.2 $\pm$ 0.3	73.4 $\pm$ 0.3	62.9 $\pm$ 0.6	72.1 $\pm$ 0.2	69.5 $\pm$ 0.3
ByT5-small*		300M	20.4 $\pm$ 0.1	74.8 $\pm$ 0.2	64.4 $\pm$ 0.2	73.1 $\pm$ 0.2	70.8 $\pm$ 0.2
ByT5-small*	+comments	300M	21.0 $\pm$ 0.3	75.0 $\pm$ 0.0	64.6 $\pm$ 0.2	73.4 $\pm$ 0.2	71.0 $\pm$ 0.1
ByT5-base*		582M	21.9 $\pm$ 0.4	75.5 $\pm$ 0.0	65.0 $\pm$ 0.2	73.8 $\pm$ 0.2	71.4 $\pm$ 0.0
ByT5-base*	+comments	582M	22.0 $\pm$ 0.1	75.5 $\pm$ 0.1	66.0 $\pm$ 0.2	73.7 $\pm$ 0.4	71.7 $\pm$ 0.1
<i>Open Source LLMs 5-Shot In-Context Learning</i>							
LLaMA-3.1-8B	retrieval-sBERT	8B	13.7	67.0	59.4	62.6	63.0
Qwen2.5-72B	retrieval-sBERT	72B	19.9	72.6	68.0	70.3	70.3
Qwen3-235B	retrieval-sBERT	235B	21.7	73.3	66.3	72.1	70.6
DeepSeek-V3	retrieval-sBERT	671B	21.9	74.6	68.3	72.6	71.8
<i>Closed Source LLMs 5-Shot In-Context Learning</i>							
GPT-4*	retrieval-sBERT	$\sim$ 1.76T	23.4	75.7	69.9	74.0	73.2
GPT-4.1	retrieval-sBERT	N/A	22.0	74.5	68.3	72.8	71.8
GPT-4o	retrieval-sBERT	$\sim$ 200B	22.8	74.7	69.1	72.4	72.1
Claude-4-sonnet	retrieval-sBERT	N/A	21.7	75.5	68.3	73.0	72.3
<i>Ours</i>							
OSMT-small		300M	20.7 $\pm$ 0.2	74.7 $\pm$ 0.4	63.8 $\pm$ 0.3	73.0 $\pm$ 0.3	70.5 $\pm$ 0.1
OSMT-small	+TRA	300M	24.6 $\pm$ 0.6	75.4 $\pm$ 0.4	73.2 $\pm$ 0.3	71.9 $\pm$ 0.4	73.5 $\pm$ 0.3
OSMT-small	+TRA +comments	300M	25.6 $\pm$ 0.8	76.4 $\pm$ 0.3	75.3$\pm$0.2	74.2 $\pm$ 0.6	75.3 $\pm$ 0.3
OSMT-base		582M	22.4 $\pm$ 0.1	75.6 $\pm$ 0.2	65.2 $\pm$ 0.8	73.7 $\pm$ 0.2	71.5 $\pm$ 0.2
OSMT-base	+TRA	582M	25.8 $\pm$ 0.1	76.5 $\pm$ 0.1	74.0 $\pm$ 0.2	72.8 $\pm$ 0.5	74.4 $\pm$ 0.2
OSMT-base	+TRA +comments	582M	26.3$\pm$0.2	77.0$\pm$0.3	75.3$\pm$0.4	74.3$\pm$0.3	75.5$\pm$0.3

elements returned by either query, thus yielding a continuous score between 0 and 1 that reflects partial correctness.

For the OverpassQL-to-Text task, we evaluate the quality of generated natural language descriptions using standard text generation metrics: BLEU-4 [26] for precision-oriented fluency assessment, ROUGE [27] for recall-based content overlap, and METEOR [28] for capturing semantic alignment with reference texts. These metrics collectively reflect the accuracy, fluency, and informativeness of the generated outputs.

C. Baselines

To evaluate the performance of OSMT, we compare it against three categories of baselines: (1) fine-tuned open-source models, (2) open-source LLMs with in-context learning, and (3) closed-source LLMs with in-context learning. This categorization reflects the broader landscape of both transparent, customizable models and proprietary commercial systems, and is applied consistently across the two tasks: *Text-to-OverpassQL* and *OverpassQL-to-Text*.

Fine-Tuned Open-Source Models. We evaluate CodeT5 [29] and CodeT5+ [30] (base), and ByT5 [21] (small, base), all fine-tuned on the OverpassNL training dataset. For the

Text-to-OverpassQL task, we follow [20] and optionally append natural language comments to each input query. For the OverpassQL-to-Text task, the models are trained using Overpass queries as input and natural language as output. Results marked with * are taken directly from [20] for fair comparison. All models are trained and evaluated over three random seeds, and we report the mean and standard deviation.

Open-Source LLMs (5-Shot In-Context Learning). We include LLaMA-3.1-8B [23], Qwen2.5-72B [31], Qwen3-235B [32], and DeepSeek-V3(0324) [33], evaluated under a 5-shot in-context learning (ICL) setup. For both tasks, demonstrations are selected from the OverpassNL training set via sBERT-based retrieval, with no parameter updates performed during inference.

Closed-Source LLMs (5-Shot In-Context Learning). To assess the upper-bound performance of proprietary systems, we evaluate GPT-4 [34], GPT-4.1(20250414), GPT-4o [35], and Claude-4-sonnet(20250514) under the same 5-shot ICL configuration. For GPT-4, we adopt the results reported in [20], while noting that version discrepancies may cause slight deviations. For the OverpassQL-to-Text task, we additionally include GPT-4o-Mini to evaluate the effectiveness of lighter-

TABLE II: Performance comparison on the Text-to-OverpassQL task using the OverpassNL test dataset. **OverpassT5** [20] refers to the ByT5-base model, finetuned on the training set augmented with the comments dataset. **TRA** denotes Tag Retrieval Augmentation. The best results are highlighted in **bold**. * means results from [20].

Model	Setting	#Params	EM $\uparrow$	chrF $\uparrow$	KVS $\uparrow$	TreeS $\uparrow$	OQS $\uparrow$
<i>Open Source LLMs 5-Shot In-Context Learning</i>							
LLaMA-3.1-8B	retrieval-sBERT	8B	12.4	65.35	57.54	60.77	61.22
Qwen2.5-72B	retrieval-sBERT	72B	17.4	71.07	66.31	69.21	68.86
Qwen3-235B	retrieval-sBERT	235B	20.3	72.4	66.6	70.9	70.0
DeepSeek-V3	retrieval-sBERT	671B	21.4	73.8	67.6	71.7	71.0
DeepSeek-V3	retrieval-sBERT+TRA	671B	21.8	74.5	71.1	72.3	72.6
<i>Closed Source LLMs 5-Shot In-Context Learning</i>							
GPT-4.1	retrieval-sBERT	N/A	20.6	73.5	68.6	71.5	71.2
GPT-4.1	retrieval-sBERT+TRA	N/A	20.7	74.1	71.7	71.7	72.5
GPT-4o	retrieval-sBERT	$\sim$ 200B	21.5	74.1	68.6	71.6	71.4
GPT-4o	retrieval-sBERT+TRA	$\sim$ 200B	21.7	74.5	71.2	71.5	72.4
Claude-4-sonnet	retrieval-sBERT	N/A	20.7	74.5	68.9	71.6	71.6
Claude-4-sonnet	retrieval-sBERT+TRA	N/A	22.1	75.1	72.1	72.5	73.2
<i>Previous SOTA</i>							
OverpassT5*	+comments	582M	20.7 $\pm$ 0.2	74.9 $\pm$ 0.1	66.1 $\pm$ 0.3	72.7 $\pm$ 0.2	71.2 $\pm$ 0.2
GPT4*	retrieval-sBERT	$\sim$ 1.76T	20.7	73.6	68.6	72.0	71.4
<i>Ours</i>							
OSMT-base		582M	20.6 $\pm$ 0.8	75.7 $\pm$ 0.5	67.6 $\pm$ 0.9	72.8 $\pm$ 0.4	71.4 $\pm$ 0.6
OSMT-base	+comments	582M	21.3 $\pm$ 0.6	75.2 $\pm$ 0.2	65.5 $\pm$ 0.4	73.2 $\pm$ 0.2	71.3 $\pm$ 0.2
OSMT-base	+TRA	582M	24.3 $\pm$ 0.1	75.8 $\pm$ 0.0	71.7 $\pm$ 0.2	71.2 $\pm$ 0.5	72.9 $\pm$ 0.3
OSMT-base	+TRA +comments	582M	24.5$\pm$1.3	76.3$\pm$0.3	73.6$\pm$0.5	73.4$\pm$0.5	74.4$\pm$0.4

TABLE III: Text-to-OverpassQL results on the OverpassNL test set, including execution metrics (EX, EX_{SOFT}).

Model	Setting	EX $\uparrow$	EX _{SOFT} $\uparrow$
<i>Best Open Source LLMs 5-Shot In-Context Learning</i>			
DeepSeek-V3	retrieval-sBERT	36.4	50.1
DeepSeek-V3	retrieval-sBERT+TRA	37.6	51.2
<i>Best Closed Source LLMs 5-Shot In-Context Learning</i>			
Claude-4-sonnet	retrieval-sBERT	37.4	52.2
Claude-4-sonnet	retrieval-sBERT+TRA	39.8	53.7
<i>Previous SOTA</i>			
OverpassT5*	+comments	33.9 $\pm$ 0.1	46.3 $\pm$ 0.3
GPT4*	retrieval-sBERT	38.9	53.0
<i>Ours</i>			
OSMT-base	+comments	35.0 $\pm$ 0.2	47.4 $\pm$ 0.3
OSMT-base	+TRA +comments	40.6$\pm$0.6	54.6$\pm$0.7

weight commercial models. Demonstrations are retrieved with the same sBERT-based strategy as in the open-source case.

V. EXPERIMENTAL RESULTS

We present and analyze the experimental results for both tasks individually: Text-to-OverpassQL in Section V-A and OverpassQL-to-Text in Section V-B. The ablation study is discussed in Section V-C, highlighting the contribution of key components within our framework.

A. Text-to-OverpassQL

1) *Main Results:* Table I presents the performance comparison on the Text-to-OverpassQL task using the OverpassNL

validation dataset. The proposed OSMT models, in both `small` and `base` configurations, consistently outperform all fine-tuned baselines and in-context learning methods across standard evaluation metrics. Among the baselines, ByT5-base (also referred to as OverpassT5 [20]) achieves an OQS score of 71.7. In contrast, our best-performing model reaches an OQS of 75.5, indicating stronger alignment with the reference queries. In addition, OSMT surpasses the performance of leading open-source and closed-source LLMs, including GPT-4 with sBERT-based retrieval (OQS 73.2), and continues to exhibit clear advantages over other high-capacity models with significantly larger parameter counts.

Table II further examines the effect of TRA in both fine-tuned and in-context learning settings on the OverpassNL test dataset. While OverpassT5 and GPT-4 with retrieval-based prompting demonstrate solid performance, they are consistently outperformed by OSMT across all metrics, supporting the overall effectiveness of our training pipeline. Although models such as DeepSeek-V3, GPT-4.1, GPT-4o, and Claude-4-sonnet benefit from TRA (as illustrated in Figure 4a), their performance remains below that of OSMT, with the largest gap observed in the KVS metric, which evaluates the accuracy of generated key-value structures. Unlike these models that apply TRA only at inference time, OSMT leverages tag-related corpora during pretraining, yielding more precise and contextually aligned tag generation. These results highlight the advantage of integrating structured tag knowledge at the

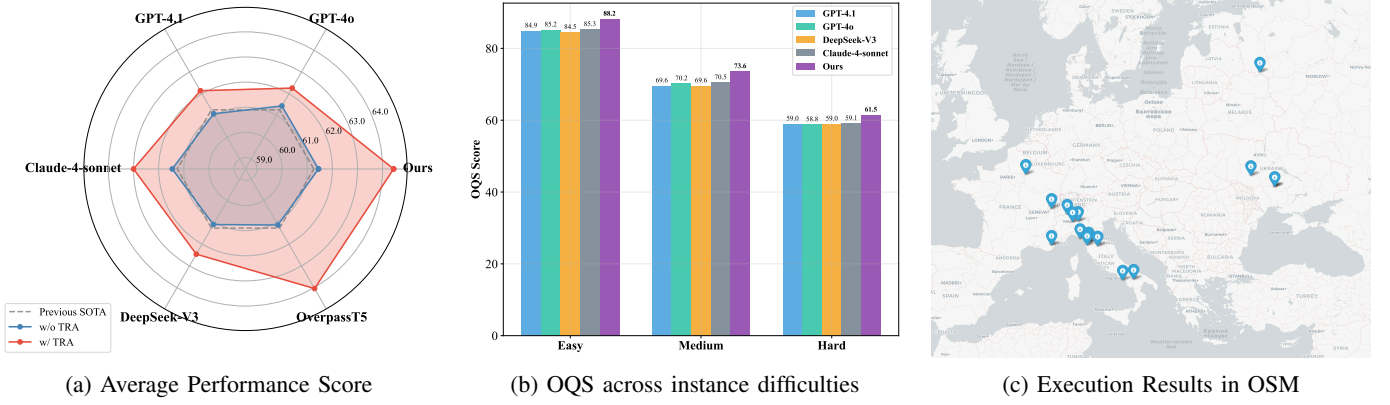


Fig. 4: Model comparison and qualitative analysis. (a) Average metric improvements from TRA across baselines. (b) OQS scores under different difficulty levels. (c) Example case showing accurate OverpassQL generation by OSMT.

architectural level and reinforce the importance of tag-aware modeling in geospatial query translation. To complement these findings with execution-level evaluation, Table III reports both EX and EX_{SOFT}. OSMT again achieves the highest scores, confirming that its improvements extend beyond syntactic and semantic similarity to the correctness of query execution.

2) *Additional Results:* To evaluate generalization under distributional shift, we divide the test set into three difficulty levels: Easy, Medium, and Hard. The partition is obtained by computing each instance’s maximum OQS similarity to the training set and then splitting the distribution into three equal-sized groups. Figure 4b shows the OQS scores across these subsets. OSMT achieves the highest performance on all levels, with 88.2 on Easy, 73.6 on Medium, and 61.5 on Hard, surpassing GPT-4.1, GPT-4o, DeepSeek-V3, and Claude-4-sonnet. The margin is relatively small for Easy queries but becomes larger for Medium and Hard cases, highlighting OSMT’s ability to maintain structural and semantic accuracy under more challenging conditions. Even on the 333 hardest instances, where all models degrade, OSMT remains the best performer, demonstrating strong robustness and generalization.

3) *Case Study:* Table IV illustrates a representative example from the Text-to-OverpassQL task, demonstrating the benefits of tag-aware modeling. The input query asks for the count of multipolygon square areas associated with unclassified highways. Baseline models typically fail on this case, either omitting critical tags such as "place"="square" or inserting irrelevant ones, which leads to incorrect results. LLMs augmented with TRA can recover the correct tag "place"="square", but often miss the "way" element, producing structurally invalid queries. In contrast, OSMT, through tag-aware pretraining combined with TRA, generates a valid OverpassQL query that faithfully captures the intent and returns the correct result, as shown in Figure 4c.

B. OverpassQL-to-Text

1) *Results:* Table V reports results for the OverpassQL-to-Text task. OSMT-base attains the highest scores across all text generation metrics (BLEU-4, ROUGE-1/2/L, and METEOR). Relative to the strongest fine-tuned baseline, ByT5-base, it

improves METEOR from 0.621 to 0.625, indicating better fluency and semantic alignment. These gains stem from pre-training on natural language inputs, OSM tag semantics, and structured OverpassQL data, which enhances the model’s ability to capture cross-modal relationships between structured queries and textual descriptions. OSMT-base also surpasses the best-performing open-source LLM, DeepSeek-V3 (METEOR 0.613), and the strongest closed-source model, GPT-4o (METEOR 0.622), despite using substantially fewer parameters. Overall, the results highlight the advantages of domain-specific pre-training and the integration of structured knowledge in geospatial text generation.

2) *Case Study:* Table VI presents an input query that counts all nodes tagged as pub, restaurant, bar, cafe, biergarten, or fast food within the current map view. Correct interpretation requires understanding both the semantic scope (relevant amenities) and the aggregation intent (the count operation). Most baseline models identify the amenities correctly but fail to express the quantitative aspect in their natural language outputs, leading to incomplete interpretations. In contrast, OSMT captures both the structural elements and the user’s intent, explicitly conveying the how many component.

C. Ablation Study

1) *Impact of Each Pre-training Objectives:* To assess the contribution of each pre-training objective, we compare three configurations: full hybrid pre-training objectives using both Masked Language Modeling and Bidirectional Translation, MLM-only, and a no-pretraining baseline. As shown in Table VII, the hybrid objective consistently outperforms the other settings across all metrics, demonstrating the importance of aligning the model with both generative and reconstructive learning signals. Removing BT leads to a measurable drop in structural and semantic accuracy, while the absence of pre-training further degrades performance, confirming the necessity of task-specific representation learning.

2) *Impact of Each Model Components:* To further isolate the contributions of individual components, we perform an ablation study on three factors: Tag Retrieval Augmentation,

TABLE IV: Case study of Text-to-OverpassQL generations from different models on a representative NL query case.

NL Query Ground Truth OverpassQL	Count of multipolygon square area with unclassified highway. [out:csv(;;count)][timeout:240];(way[“area”=“yes”][“highway”=“unclassified”][“place”=“square”](if:is_closed());relation[“type”=“multipolygon”][“highway”=“unclassified”][“place”=“square”];);out count;→ count = 15 as shown in Figure 4c
OverpassT5 (✗)	[out:json][timeout:25];(way[“highway”=“unclassified”][“surface”=“unclassified”];relation[“type”=“multipolygon”][“highway”=“unclassified”][“surface”=“unclassified”];);out count;→ Incorrect Result: count = 0
Qwen3-235B (✗)	[out:csv(;;count)][timeout:250];(way[“area”=“yes”][“highway”=“unclassified”](if:is_closed());relation[“type”=“multipolygon”][“highway”=“unclassified”];);out count;→ Incorrect Result: count = 1776
DeepSeek-V3 (✗)	[out:csv(;;count)][timeout:240];(relation[“type”=“multipolygon”][“highway”=“unclassified”][“area”=“square”];);out count;→ Incorrect Result: count = 0
GPT-4 (✗)	[out:csv(;;count)][timeout:240];(relation[“type”=“multipolygon”][“highway”=“unclassified”][“area”=“yes”];);out count;→ Incorrect Result: count = 37
GPT-4.1 (✗)	[out:csv(;;count)][timeout:240];relation[“type”=“multipolygon”][“highway”=“unclassified”][“area”=“yes”][“shape”=“square”];out count;→ Incorrect Result: count = 0
GPT-4o (✗)	[out:csv(;;count)][timeout:240];relation[“type”=“multipolygon”][“highway”=“unclassified”][“area”=“yes”][“shape”=“square”];out count;→ Incorrect Result: count = 0
Claude-4-sonnet (✗)	[out:csv(;;count)][timeout:240];(relation[“type”=“multipolygon”][“highway”=“unclassified”][“area”=“yes”];);out count;→ Incorrect Result: count = 37
DeepSeek-V3 + TRA (✗)	[out:csv(;;count)][timeout:240];(relation[“type”=“multipolygon”][“place”=“square”][“highway”=“unclassified”][“area”=“yes”];);out count;→ Incorrect Result: count = 1
Claude-4-sonnet + TRA (✗)	[out:csv(;;count)][timeout:240];(relation[“type”=“multipolygon”][“place”=“square”][“highway”=“unclassified”][“area”=“yes”];);out count;→ Incorrect Result: count = 1
Ours + TRA (✓)	[out:csv(;;count)][timeout:240];(way[“area”=“yes”][“highway”=“unclassified”][“place”=“square”](if:is_closed());relation[“type”=“multipolygon”][“highway”=“unclassified”][“place”=“square”];);out count;→ Correct Result: count = 15

TABLE V: Comparative performance analysis for OverpassQL-to-Text task. The best results are highlighted in **bold**.

Method	Setting	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑
<i>Finetuning</i>						
CodeT5-small		0.355±0.006	0.639±0.008	0.461±0.009	0.611±0.007	0.603±0.004
CodeT5-base		0.358±0.010	0.636±0.002	0.455±0.006	0.606±0.003	0.605±0.006
ByT5-small		0.362±0.009	0.645±0.003	0.473±0.005	0.618±0.005	0.609±0.004
ByT5-base		0.362±0.004	0.647±0.004	0.467±0.004	0.618±0.004	0.621±0.004
<i>Open Source LLMs 5-Shot In-Context Learning</i>						
LLaMA-3.1-8B	retrieval-sBERT	0.202	0.609	0.416	0.572	0.587
Qwen2.5-72B	retrieval-sBERT	0.209	0.614	0.425	0.577	0.611
DeepSeek-V3	retrieval-sBERT	0.201	0.610	0.414	0.568	0.613
<i>Closed Source LLMs 5-Shot In-Context Learning</i>						
GPT-4o-Mini	retrieval-sBERT	0.213	0.627	0.433	0.585	0.613
GPT-4.1	retrieval-sBERT	0.226	0.626	0.436	0.585	0.618
GPT-4o	retrieval-sBERT	0.231	0.636	0.446	0.596	0.622
Claude-4-sonnet	retrieval-sBERT	0.212	0.604	0.419	0.572	0.599
<i>Ours</i>						
OSMT-small		0.369±0.003	0.651±0.002	0.475±0.003	0.625±0.004	0.618±0.000
OSMT-base		0.372±0.002	0.655±0.002	0.480±0.001	0.627±0.002	0.625±0.004

TABLE VI: Case study of OverpassQL-to-Text generations from different models on a representative OverpassQL case.

OverpassQL	[out:csv(;;id,“name”,“amenity”,“cuisine”,“addr:street”,“addr:housenumber”,“opening_hours”,“contact:phone”,“contact:email”,“contact:website”;;lat,lon,brand,“count:true”,“”);(node[“amenity”=“pub”]({{bbox}});node[“amenity”=“restaurant”]({{bbox}});node[“amenity”=“bar”]({{bbox}});node[“amenity”=“cafe”]({{bbox}});node[“amenity”=“biertgarten”]({{bbox}});node[“amenity”=“fast_food”]({{bbox}}););out; out count; Nodes tagged as a pub, restaurant, bar, cafe, biertgarten or fast food restaurant in the current view and how many they are.
Ground Truth Description	
ByT5-base (✗)	Residential land in the current view.
LLaMA-3.1-8B (✗)	Pubs, restaurants, bars, cafes, biertgardens and fast food restaurants in the selected window.
GPT-4o (✗)	Pubs, restaurants, bars, cafes, biertgardens, and fast food locations in the current view with detailed contact information.
Qwen2.5-72B (✗)	Pubs, restaurants, bars, cafes, biertgardens, and fast food places in the selected window, including detailed information such as name, amenity type, cuisine, address, opening hours, and contact details.
DeepSeek-V3 (✗)	Pubs, restaurants, bars, cafes, biertgardens, and fast food establishments in the selected window, including details such as name, cuisine, address, opening hours, contact information, and brand.
GPT-4o-mini (✗)	Pubs, restaurants, bars, cafes, biertgardens, and fast food locations in the selected window, with detailed information including name, cuisine, address, opening hours, and contact details, in CSV format.
GPT-4.1 (✗)	Pubs, restaurants, bars, cafes, biertgardens, and fast food places in the selected area, including their names, amenities, cuisine, addresses, opening hours, contact information, brand, location.
Claude-4-sonnet (✗)	Pubs, restaurants, bars, cafes, biertgardens and fast food establishments in the selected area with detailed contact and location information.
Ours (✓)	Pubs, restaurants, bars, cafes, biertgardens and fast food restaurants in the current view and how many there are.

TABLE VII: Ablation study comparing hybrid pre-training (MLM+BT), MLM-only, and no-pretraining for both Text-to-OverpassQL and OverpassQL-to-Text. All models share the same architecture and training setup, and results are averaged over three random seeds. Results with best scores in **bold**.

MLM	BT	EM $\uparrow$	OQS $\uparrow$	BLEU-4 $\uparrow$	ROUGE-L $\uparrow$	METEOR $\uparrow$
✓	✓	20.6$\pm$0.8	71.4$\pm$0.6	0.372$\pm$0.002	0.627$\pm$0.002	0.625$\pm$0.004
✓	–	20.4 $\pm$ 0.7	70.9 $\pm$ 0.7	0.365 $\pm$ 0.008	0.621 $\pm$ 0.002	0.625$\pm$0.001
–	–	19.7 $\pm$ 0.9	70.1 $\pm$ 0.4	0.362 $\pm$ 0.004	0.618 $\pm$ 0.004	0.621 $\pm$ 0.004

training data augmented with the comments dataset, and pre-training. As shown in Table VIII, each component yields measurable performance improvements. Removing TRA results in a significant decline in KVS and OQS scores, underscoring its importance for structural alignment. Excluding the comments dataset also degrades performance, indicating its usefulness in clarifying query semantics. The use of ground-truth tags in TRA establishes an upper bound, suggesting additional gains are achievable through improved tag retrieval. Overall, these findings validate the architectural design of OSMT and highlight the complementary benefits of its core components.

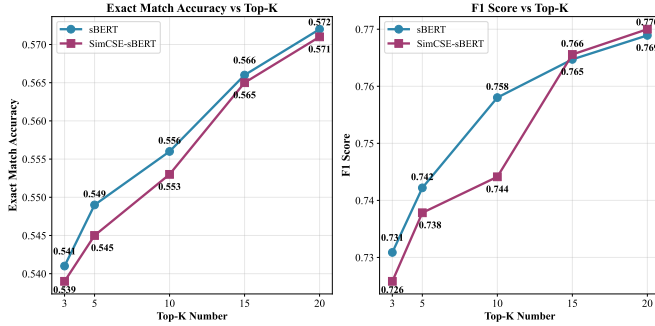


Fig. 5: Ablation study on TRA: comparison of embedding models (sBERT vs. SimCSE-sBERT) across different Top- k settings, evaluated by Exact Match Accuracy and F1 Score.

3) *Impact of Each TRA Components*: We further examine the effect of embedding models within the TRA framework by comparing sBERT and SimCSE-sBERT under varying Top- k retrieval settings. Figure 5 reports Exact Match and F1 scores against ground-truth tag sets, directly measuring retrieval quality. The results, further validated in Table IX, show complementary strengths: sBERT yields slightly higher EM at larger k , while SimCSE-sBERT provides more stable F1 improvements, especially with comments. Based on the consistently superior F1 performance of SimCSE-sBERT on the development set, we adopt the 20-shot configuration as the final TRA setting in our experiments.

VI. DISCUSSION

A. Generalization Across Geo-Related Datasets

We primarily focused on OverpassNL since it reflects realistic usage of **OverpassQL**, the production query language of OSM that supports global coverage and directly exercises the full richness of the tag system. To test generality, we also fine-tune **OSMT-small** on *GeoQuery* [36] and *NLmaps* v2 [37]. As shown in Table X, explicit linking strategies consistently

improve performance: for *GeoQuery*, we adopt an analogous schema-linking variant of TRA, while for *NLmaps* v2, TRA applies directly on OSM tags. In both cases, our approach surpasses **ByT5-small**, demonstrating that (i) OSMT is not confined to OverpassQL and (ii) Tag/schema-aware mechanisms provide robust gains across structured query languages.

B. Relation of TRA to Mainstream RAG Methods

Conceptually, TRA follows the same paradigm as Retrieval-Augmented Generation (RAG): retrieve external knowledge at inference time and condition generation on the retrieved signals. The difference lies in its specialization to the OSM/OverpassQL setting. First, instead of unstructured passages, TRA retrieves symbolic OSM tags from a curated tag knowledge base $\mathcal{K}$. Second, $\mathcal{K}$ is constructed not from a static corpus but by seeding observed query-tag pairs and synthesizing coverage for the entire official tag set, ensuring completeness beyond training data. Third, instead of appending long passages, TRA supplies refined and validated canonical tags as control signals, ensuring concise conditioning and strict compliance with OverpassQL’s syntax and ontology.

VII. RELATED WORK

A. Natural Language to Structured Query Generation

Translating natural language into structured query languages has long been a fundamental challenge in the database community [1], [3], [6], [7], [12], [38], [39], as reflected in the extensively studied Text-to-SQL and Text-to-Vis tasks. This line of research is pivotal for enabling intuitive access to data, allowing users to articulate complex information needs without requiring expertise in formal query syntax. While substantial advances have been made in the context of traditional relational databases, structured geospatial databases pose additional challenges due to their hierarchical organization, compositional semantics, and topologically enriched schema structures. An early effort in this direction is *GeoQuery* [36], [40], a logical query language developed for a small U.S. geography database, accompanied by a manually constructed corpus of natural language questions. Building on this foundation, later work introduced grammar induction techniques to map natural language to logical forms, along with probabilistic models for ranking competing parses [41]. Subsequently, *NLmaps* [42] and its extended version *NLmaps* v2 [37] were proposed, employing a machine-readable language (MRL) that abstracts OverpassQL, albeit with support for only a limited subset of its expressive features. *NLmaps* v2 expanded coverage by incorporating a broader set of natural language questions and OSM tags. More recently, *OverpassNL* [20] was introduced as a dataset comprising natural language questions collected from Overpass Turbo⁴, each paired with an OverpassQL query authored by OSM community members. This resource facilitated the definition of the Text-to-OverpassQL task, which aims to directly translate natural language into executable OverpassQL queries without relying on intermediate representations or abstractions.

⁴<https://overpass-turbo.eu/>

TABLE VIII: Ablation study on the model components. **TRA** indicates Tag Retrieval Augmentation and **PT** indicates pre-training. The best results are shown in **bold**. The gray row uses ground-truth tags for TRA as an upper bound.

Setting	PT	TRA	Comments	EM $\uparrow$	chrF $\uparrow$	KVS $\uparrow$	TreeS $\uparrow$	OQS $\uparrow$
Upper bound	✓	✓	✓	29.0 $\pm$ 0.4	84.5 $\pm$ 0.4	97.0 $\pm$ 0.2	77.9 $\pm$ 0.3	86.4 $\pm$ 0.2
Ours	✓	✓	✓	24.5$\pm$1.3	76.3$\pm$0.3	73.6$\pm$0.5	73.4$\pm$0.5	74.4$\pm$0.4
w/o Comments	✓	✓	–	24.3 $\pm$ 0.1	75.8 $\pm$ 0.0	71.7 $\pm$ 0.2	71.2 $\pm$ 0.5	72.9 $\pm$ 0.3
w/o TRA	✓	–	✓	21.3 $\pm$ 0.6	75.2 $\pm$ 0.2	65.5 $\pm$ 0.4	73.2 $\pm$ 0.2	71.3 $\pm$ 0.2
w/o TRA & Comments	✓	–	–	20.6 $\pm$ 0.8	75.7 $\pm$ 0.5	67.6 $\pm$ 0.9	72.8 $\pm$ 0.4	71.4 $\pm$ 0.6
w/o PT	–	✓	✓	24.0 $\pm$ 0.9	76.2 $\pm$ 0.4	73.4 $\pm$ 0.3	73.3 $\pm$ 0.2	74.3 $\pm$ 0.3
w/o PT & Comments	–	✓	–	23.7 $\pm$ 0.2	75.4 $\pm$ 0.1	71.2 $\pm$ 0.1	71.0 $\pm$ 0.3	72.5 $\pm$ 0.2
w/o PT & TRA	–	–	✓	20.7 $\pm$ 0.2	74.9 $\pm$ 0.1	66.1 $\pm$ 0.3	72.7 $\pm$ 0.2	71.2 $\pm$ 0.2
w/o PT & TRA & Comments	–	–	–	19.7 $\pm$ 0.9	74.4 $\pm$ 0.3	63.4 $\pm$ 0.8	72.4 $\pm$ 0.2	70.1 $\pm$ 0.4

TABLE IX: Ablation studies on TRA: effect of embedding models and retrieval Top-k. Best results are shown in **bold**.

Setting	EM $\uparrow$	chrF $\uparrow$	KVS $\uparrow$	TreeS $\uparrow$	OQS $\uparrow$
<i>Top-5 TRA with sBERT</i>					
+TRA	22.1 $\pm$ 0.9	76.3$\pm$0.1	72.3 $\pm$ 0.2	73.7 $\pm$ 0.3	74.1 $\pm$ 0.2
+TRA +comments	22.5 $\pm$ 0.7	76.3$\pm$0.1	72.4 $\pm$ 0.2	74.0$\pm$0.4	74.2 $\pm$ 0.1
<i>Top-20 TRA with sBERT</i>					
+TRA	24.2 $\pm$ 0.2	75.7 $\pm$ 0.1	71.1 $\pm$ 0.2	70.8 $\pm$ 0.4	72.6 $\pm$ 0.2
+TRA +comments	24.6$\pm$1.0	76.3$\pm$0.3	73.2 $\pm$ 0.3	73.6 $\pm$ 0.6	74.4$\pm$0.3
<i>Top-20 TRA with SimCSE-sBERT</i>					
+TRA	24.3 $\pm$ 0.1	75.8 $\pm$ 0.0	71.7 $\pm$ 0.2	71.2 $\pm$ 0.5	72.9 $\pm$ 0.3
+TRA +comments	24.5 $\pm$ 1.3	76.3$\pm$0.3	73.6$\pm$0.5	73.4 $\pm$ 0.5	74.4$\pm$0.4

TABLE X: Generalization performance on GeoQuery and NLmaps v2 datasets. Best results are shown in **bold**.

Model	Setting	GeoQuery		NLmaps V2	
		EM	F1 Score	EM	F1 Score
<i>Baseline</i>					
ByT5-small		49.76±2.15	80.97±0.47	73.83±1.87	77.66±1.37
ByT5-small	+TRA	68.81±0.90	90.52±0.32	91.12±0.98	92.42±0.72
<i>Ours</i>					
OSMT-small		49.64±1.43	81.41±0.65	74.73±1.80	78.47±1.41
OSMT-small	+TRA	69.64±0.95	90.68±0.93	91.23±0.23	92.44±0.08

B. Structured Query Explanation

Generating natural language from structured queries improves the interpretability and usability of data-centric systems by explaining query semantics and intent without requiring expertise in the underlying language. Most prior work has focused on SQL-to-text translation [43]–[49], typically evaluated on benchmarks such as WikiSQL [50] and Spider [51]. These studies, however, remain confined to the relational paradigm. In contrast, our work targets OSM, a community-maintained geospatial database with a node–way–relation model and a rich tagging system. Its query language, OverpassQL, poses unique challenges due to recursive syntax, hierarchical structures, and topological reasoning. We therefore formalize the OverpassQL-to-Text task, which translates OverpassQL queries into natural language to enhance interpretability, support transparency in geospatial retrieval, and complement Text-to-OverpassQL, together forming a bidirectional interface between natural language and structured geospatial queries.

C. Domain-Specific Language Models

Domain-specific language models build on the success of general-purpose pre-training [52]–[56], extending it to special-

ized settings such as code and structured queries. CodeT5 [29] jointly models natural and programming languages, while UNIFIEDSKG [57] pre-trains on structured grounding tasks to support semantic parsing and QA. In scientific domains, models like MolT5 [58], BioT5 [59], and BioT5+ [60] integrate specialized text with domain structures for bioinformatics. For data interfaces, DataVisT5 [3] encodes text and visualizations, and CodeS [4] provides open-source models optimized for text-to-SQL. These efforts highlight the effectiveness of domain-specific pre-training in capturing the contextual and structural properties of specialized modalities.

VIII. CONCLUSION

In this work, we presented **OSMT**, the first open-source tag-aware pre-trained language model specifically designed to bridge the gap between natural language and OverpassQL, a structured query language for accessing OpenStreetMap data. By integrating natural language text, structured OverpassQL syntax, and OSM tag semantics through a hybrid pre-training strategy, OSMT enables accurate and structurally valid bidirectional translation between natural language and OverpassQL. Moreover, to better capture the hierarchical and relational dependencies in OSM, we introduced a Tag Retrieval Augmentation mechanism that significantly enhances semantic alignment and query robustness in the Text-to-OverpassQL task. Extensive experiments on both forward and reverse tasks demonstrate that OSMT consistently outperforms strong baselines in accuracy, interpretability, and parameter efficiency. These findings highlight the effectiveness of cross-representational pre-training for geospatial query tasks and position OSMT as a solid foundation for future research on natural language interfaces to structured geospatial databases.

ACKNOWLEDGMENT

Yuanfeng Song is the corresponding author. The work described in this paper was partially supported by grants from : 1) the NSFC/RGC Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the National Natural Science Foundation of China (Project No. N_PolyU5179/25); 2) the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU25600624); 3) the Innovation Technology Fund (Project No. ITS/052/23MX, PRP/009/22FX, and PRP/004/25FX).

AL-GENERATED CONTENT ACKNOWLEDGMENT

The development of ideas, algorithms, and experiments in this work was completed entirely by the authors. Large language models were used only lightly to polish the writing, such as fixing typos and improving grammar.

REFERENCES

- [1] T. Ren, Y. Fan, Z. He, R. Huang, J. Dai, C. Huang, Y. Jing, K. Zhang, Y. Yang, and X. S. Wang, "PURPLE: Making a Large Language Model a Better SQL Writer," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2024, pp. 15–28. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/ICDE60146.2024.00009>
- [2] Y. Fan, Z. He, T. Ren, C. Huang, Y. Jing, K. Zhang, and X. S. Wang, "Metasql: A Generate-Then-Rank Framework for Natural Language to SQL Translation," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2024, pp. 1765–1778. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/ICDE60146.2024.00143>
- [3] Z. Wan, Y. Song, S. Li, C. J. Zhang, and R. C.-W. Wong, "DataVisT5: A Pre-Trained Language Model for Jointly Understanding Text and Data Visualization," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2025, pp. 1704–1717. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/ICDE65448.2025.00131>
- [4] H. Li, J. Zhang, H. Liu, J. Fan, X. Zhang, J. Zhu, R. Wei, H. Pan, C. Li, and H. Chen, "Codes: Towards building open-source language models for text-to-sql," *Proc. ACM Manag. Data*, vol. 2, no. 3, May 2024. [Online]. Available: <https://doi.org/10.1145/3654930>
- [5] Z. Gu, J. Fan, N. Tang, L. Cao, B. Jia, S. Madden, and X. Du, "Few-shot text-to-sql translation using structure and content prompt learning," *Proc. ACM Manag. Data*, vol. 1, no. 2, Jun. 2023. [Online]. Available: <https://doi.org/10.1145/3589292>
- [6] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1235–1247.
- [7] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang, "The dawn of natural language to sql: Are we fully ready?" *Proc. VLDB Endow.*, vol. 17, no. 11, p. 3318–3331, Jul. 2024. [Online]. Available: <https://doi.org/10.14778/3681954.3682003>
- [8] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou, "Text-to-sql empowered by large language models: A benchmark evaluation," *Proc. VLDB Endow.*, vol. 17, no. 5, p. 1132–1145, Jan. 2024. [Online]. Available: <https://doi.org/10.14778/3641204.3641221>
- [9] S. Li, Y. Song, X. Chen, A. Peng, Z. Wan, C. J. Zhang, and R. C.-W. Wong, "Mind your questions! towards backdoor attacks on text-to-visualization models," 2024. [Online]. Available: <https://arxiv.org/abs/2410.06782>
- [10] H. Fu, C. Liu, B. Wu, F. Li, J. Tan, and J. Sun, "Catsql: Towards real world natural language to sql applications," *Proc. VLDB Endow.*, vol. 16, no. 6, p. 1534–1547, Feb. 2023. [Online]. Available: <https://doi.org/10.14778/3583140.3583165>
- [11] W. Zhang, Y. Wang, Y. Song, V. J. Wei, Y. Tian, Y. Qi, J. H. Chan, R. C.-W. Wong, and H. Yang, "Natural language interfaces for tabular data querying and visualization: A survey," *IEEE Trans. on Knowl. and Data Eng.*, vol. 36, no. 11, p. 6699–6718, Nov. 2024. [Online]. Available: <https://doi.org/10.1109/TKDE.2024.3400824>
- [12] S. Li, X. Chen, Y. Song, Y. Song, C. J. Zhang, F. Hao, and L. Chen, "Prompt4Vis: prompting large language models with example mining for tabular data visualization," *The VLDB Journal*, vol. 34, no. 38, 2025. [Online]. Available: <https://doi.org/10.1007/s00778-025-00912-0>
- [13] G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu, C. Cundy, Z. Li, R. Zhu, and N. Lao, "On the opportunities and challenges of foundation models for geospatial artificial intelligence," 2023. [Online]. Available: <https://arxiv.org/abs/2304.06798>
- [14] C. Deng, T. Zhang, Z. He, Y. Xu, Q. Chen, Y. Shi, L. Fu, W. Zhang, X. Wang, C. Zhou, Z. Lin, and J. He, "K2: A foundation language model for geoscience knowledge understanding and utilization," 2023. [Online]. Available: <https://arxiv.org/abs/2306.05064>
- [15] R. Manvi, S. Khanna, G. Mai, M. Burke, D. Lobell, and S. Ermon, "Geollm: Extracting geospatial knowledge from large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2310.06213>
- [16] Y. Luo, Z. Wan, Y. Chen, G. Mai, F. lai Chung, and K. Larson, "Transflower: An explainable transformer-based model with flow-to-flow attention for commuting flow prediction," 2024. [Online]. Available: <https://arxiv.org/abs/2402.15398>
- [17] J. He, T. Nie, and W. Ma, "Geolocation representation from large language models are generic enhancers for spatio-temporal learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17094–17104.
- [18] H. Zhang, X. Zhu, X. Wang, H. Gu, Y. Song, and L. Ma, "Natural language to overpass query: A multi-step approach using task decomposition and key-value correction," in *Proceedings of the 26th IEEE International Conference on Mobile Data Management (MDM)*. Pittsburgh, PA, USA: IEEE, June 2025.
- [19] Y. Luo, H. Duan, Y. Liu, and F. lai Chung, "Timestamps as prompts for geography-aware location recommendation," 2023. [Online]. Available: <https://arxiv.org/abs/2304.04151>
- [20] M. Staniek, R. Schumann, M. Züfle, and S. Riezler, "Text-to-overpassql: A natural language interface for complex geodata querying of openstreetmap," *Transactions of the Association for Computational Linguistics*, vol. 12, p. 562–575, 2024. [Online]. Available: http://dx.doi.org/10.1162/tacl_a_00654
- [21] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "ByT5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022. [Online]. Available: <https://aclanthology.org/2022.tacl-1.17>
- [22] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- [23] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, and A. K. et al., "The llama 3 herd of models," 2024. [Online]. Available: <https://arxiv.org/abs/2407.21783>
- [24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: <https://aclanthology.org/D19-1410/>
- [25] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, and P. Pecina, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 392–395. [Online]. Available: <https://aclanthology.org/W15-3049>
- [26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [27] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.
- [28] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.
- [29] Y. Wang, W. Wang, S. Joty, and S. C. Hoi, "Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation," *arXiv preprint arXiv:2109.00859*, 2021.
- [30] Y. Wang, H. Le, A. D. Gotmare, N. D. Bui, J. Li, and S. C. Hoi, "Codet5+: Open code large language models for code understanding and generation," *arXiv preprint arXiv:2305.07922*, 2023.
- [31] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.
- [32] Qwen, "Qwen3 technical report," 2025. [Online]. Available: <https://arxiv.org/abs/2505.09388>
- [33] DeepSeek-AI, "Deepseek-v3 technical report," 2024. [Online]. Available: <https://arxiv.org/abs/2412.19437>

- [34] OpenAI, “Gpt-4 technical report,” 2024.
- [35] OpenAI, “Gpt-4o system card,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.21276>
- [36] J. M. Zelle and R. J. Mooney, “Learning to parse database queries using inductive logic programming,” in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, ser. AAAI’96. AAAI Press, 1996, p. 1050–1055.
- [37] C. Lawrence and S. Riezler, “Improving a neural semantic parser by counterfactual learning from human bandit feedback,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1820–1830. [Online]. Available: <https://aclanthology.org/P18-1169>
- [38] X. Liu, S. Shen, B. Li, P. Ma, R. Jiang, Y. Zhang, J. Fan, G. Li, N. Tang, and Y. Luo, “A survey of nl2sql with large language models: Where are we, and where are we going?” 2025. [Online]. Available: <https://arxiv.org/abs/2408.05109>
- [39] Z. Hong, Z. Yuan, Q. Zhang, H. Chen, J. Dong, F. Huang, and X. Huang, “Next-generation database interfaces: A survey of llm-based text-to-sql,” 2025. [Online]. Available: <https://arxiv.org/abs/2406.08426>
- [40] R. Kate, Y. Wong, and R. Mooney, “Learning to transform natural to formal languages,” in *Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP)*, vol. 3, 2005, pp. 1062–1068.
- [41] L. S. Zettlemoyer and M. Collins, “Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars,” in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI’05)*. Arlington, Virginia, USA: AUAI Press, 2005, pp. 658–666.
- [42] C. Lawrence and S. Riezler, “NLmaps: A natural language interface to query OpenStreetMap,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, H. Watanabe, Ed. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 6–10. [Online]. Available: <https://aclanthology.org/C16-2002>
- [43] G. Koutrika, A. Simitsis, and Y. E. Ioannidis, “Explaining structured queries in natural language,” in *Proceedings of the 2010 IEEE 26th International Conference on Data Engineering (ICDE)*. IEEE, 2010, pp. 333–344.
- [44] A.-C. N. Ngomo, L. Bühmann, C. Unger, J. Lehmann, and D. Gerber, “Sorry, i don’t speak SPARQL: translating SPARQL queries into natural language,” in *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013, pp. 977–988.
- [45] S. Iyer, I. Konstantas, A. Cheung, and L. Zettlemoyer, “Summarizing source code using a neural attention model,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 2073–2083.
- [46] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [47] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Long Beach, CA, USA, 2017, pp. 1025–1035.
- [48] L. Song, Y. Zhang, Z. Wang, and D. Gildea, “A graph-to-sequence model for AMR-to-text generation,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1616–1626. [Online]. Available: <https://aclanthology.org/P18-1150>
- [49] L. Wu, I. E.-H. Yen, F. Xu, P. Ravikumar, and M. Witbrock, “D2ke: From distance to kernel and embedding,” *arXiv preprint arXiv:1802.04956*, 2018.
- [50] V. Zhong, C. Xiong, and R. Socher, “Seq2sql: Generating structured queries from natural language using reinforcement learning,” *CoRR*, vol. abs/1709.00103, 2017. [Online]. Available: <http://arxiv.org/abs/1709.00103>
- [51] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman *et al.*, “Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3911–3921.
- [52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL*, 2019.
- [53] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *arXiv preprint arXiv:1910.10683*, 2019.
- [54] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [55] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, “Ernie: Enhanced representation through knowledge integration,” *arXiv preprint arXiv:1904.09223*, 2019.
- [56] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” in *ICLR*, 2020. [Online]. Available: <https://openreview.net/pdf?id=r1xMH1BtvB>
- [57] T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C.-S. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu, “Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models,” *EMNLP*, 2022.
- [58] C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, and H. Ji, “Translation between molecules and natural language,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 375–413. [Online]. Available: <https://aclanthology.org/2022.emnlp-main.26>
- [59] Q. Pei, W. Zhang, J. Zhu, K. Wu, K. Gao, L. Wu, Y. Xia, and R. Yan, “BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1102–1123. [Online]. Available: <https://aclanthology.org/2023.emnlp-main.70>
- [60] Q. Pei, L. Wu, K. Gao, X. Liang, Y. Fang, J. Zhu, S. Xie, T. Qin, and R. Yan, “Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning,” *arXiv preprint arXiv:2402.17810*, 2024.