

GeoToken: Hierarchical Geolocalization of Images via Next Token Prediction

Narges Ghasemi^{*†}, Amir Ziashahabi^{*‡}, Salman Avestimehr[‡], Cyrus Shahabi[†]

[†]*Department of Computer Science, University of Southern California, Los Angeles, CA, USA*

[‡]*Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA*
{nghasemi, ziashaha, avestime, shahabi}@usc.edu

Abstract—

Image geolocalization—the task of determining an image’s geographic origin—poses significant challenges, largely due to visual similarities across disparate locations and the large search space. To address these issues, we propose a hierarchical sequence prediction approach inspired by how humans narrow down locations from broad regions (e.g., country) to specific addresses (e.g., street name and house number). Analogously, our model predicts geographic tokens hierarchically, first identifying a general region and then sequentially refining predictions to increasingly precise locations. Rather than relying on explicit semantic partitions (e.g., country, city), our method uses S2 cells, a nested, multiresolution global grid, and sequentially predicts finer-level cells conditioned on visual inputs and previous predictions. This procedure mirrors autoregressive text generation in large language models. Much like in language modeling, final performance depends not only on training but also on inference-time strategy. We investigate multiple top-down traversal methods for autoregressive sampling, incorporating techniques from test-time compute scaling used in language models. Specifically, we integrate beam search and multi-sample inference while exploring various selection strategies to determine the final output. This approach enables the model to manage uncertainty by exploring multiple plausible paths through the hierarchy. We evaluate our method on the Im2GPS3k and YFCC4k datasets against two distinct sets of baselines: those that operate without a Multimodal Large Language Model (MLLM) and those that leverage one. In the MLLM-free setting, our model surpasses other comparable baselines on nearly all metrics, achieving state-of-the-art performance with accuracy gains of up to 13.9%. When augmented with an MLLM, our model again outperforms all baselines, setting a new state of the art across every metric. The source code is available at <https://github.com/NNargesNN/GeoToken>.

Index Terms—Image Geolocalization, Autoregressive Models, Multimodal Large Language Models, Retrieval Augmented Generation

I. INTRODUCTION

Accurately estimating the geographic coordinates where a photograph was taken, a task known as worldwide image *geolocalization*, is a long-standing problem in computer vision with broad practical applications. The ability to accurately geolocate images is crucial for organizing vast collections of visual data, enabling location-aware services in mobile applications, facilitating automatic photo organization, supporting environmental monitoring by connecting imagery to

specific places for analysis, and enhancing search capabilities by allowing users to find photos taken in a particular area. Despite its utility, achieving accurate and robust worldwide image geolocalization remains a significant challenge.

There are two main challenges underlying this problem. Firstly, visual cues indicative of location are often subtle, ambiguous, and easily confounded. Features like architectural styles, vegetation types, or even road signs might offer hints, but similar elements can appear in different parts of the world, leading to potential confusion. Secondly, geotagged imagery, as a corpus for training or retrieval, is distributed unevenly across the globe. Figure 1(a) shows the distribution of the location of the images in one of the largest datasets used for training, MP-16 [1] with over 4 million data points. Popular tourist destinations and urban centers are often densely covered, while vast rural or remote areas have sparse or no associated geotagged images. This severe data imbalance means that models trained on existing datasets are heavily biased towards well-represented locations. This can lead to models that perform well in familiar areas but struggle to generalize and accurately predict locations in uncovered or underrepresented regions.

Previous approaches to worldwide image geolocalization can be broadly categorized into three groups: classification, retrieval, and recent hybrid methods. Classification-based methods [2]–[5] partition the Earth’s surface into discrete geographic cells and train a model to predict which cell an image belongs to. While hierarchical structures were introduced to improve resolution [6], a key limitation is their reliance on a relatively small number of predefined geographic cells; the model can only output one of the few discrete classes it was trained on, making it difficult to predict precise locations that are far from the center of the cells [7]. Another fundamental strategy is image retrieval, where a query image is matched against a large database of geotagged images to find visually similar examples with known locations [6]. While effective for landmark-heavy datasets or areas with dense image coverage, retrieval struggles with the sheer scale of the Earth and the diversity of possible scenes; building a comprehensive global database is impractical, and many images, particularly in less photographed areas, will not have a close visual match [8]. Furthermore, retrieval typically provides little sense of prediction confidence or uncertainty; it’s difficult to gauge

* These authors contributed equally.

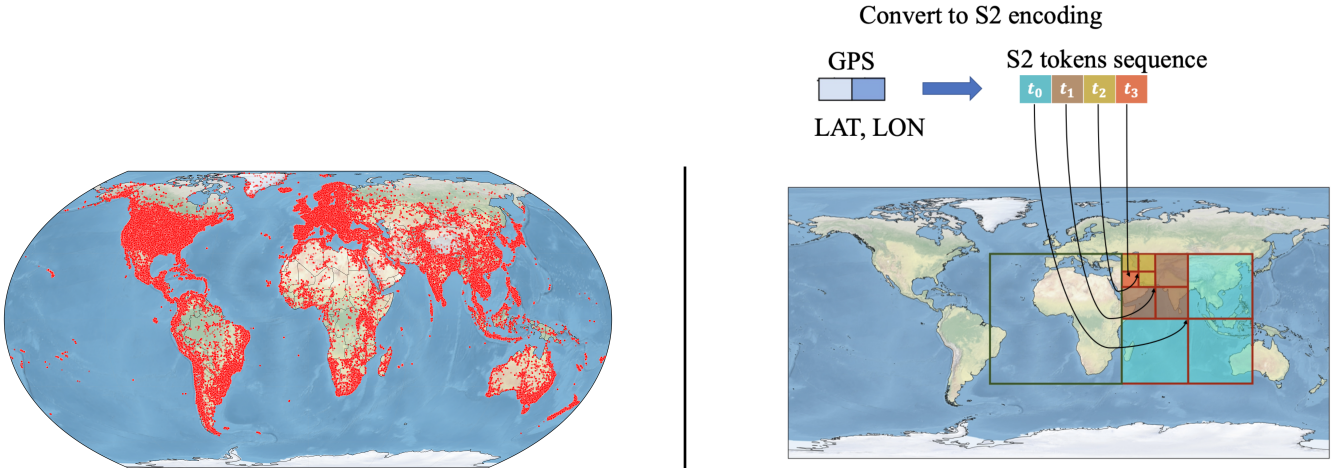


Fig. 1: (a) Distribution of the MP16 dataset, with over 4 million samples across the world; (b) Visualization of our S2 tokens.

how reliable a retrieved location is, or if a visually similar match is truly geographically accurate or merely coincidental. More recently, hybrid methods and approaches leveraging Multimodal Large Language Models (MLLMs) or contrastive learning have emerged [7], [9]–[11]. These methods often combine aspects of retrieval and generation or learn powerful embeddings. While achieving impressive results, some heavily rely on closed-source models [9], [10], can involve complex pipelines [11], and often lack an intuitive mechanism for managing prediction uncertainty.

In this paper, we introduce GeoToken, a novel approach that unifies the strengths of these diverse paradigms within a single, end-to-end framework, inspired by the coarse-to-fine reasoning process human experts employ when localizing an unfamiliar scene. Consider how a human might identify the location of a photo: they might first recognize broad regional cues (“This looks like the United States”), then refine their hypothesis based on more specific details (“The architecture suggests New York”), and finally pinpoint the city or street (“This building is clearly in Manhattan”). This process involves forming a broad initial hypothesis and progressively refining it as more evidence is considered.

Our core innovation is to translate this intuitive human strategy into a computational model by treating worldwide image geolocation as a coarse-to-fine token prediction task. Analogous to how large language models [12]–[15] generate text, one token at a time. We capture this intuition by decomposing any geographic coordinate into a sequence of hierarchical tokens, where each token represents a progressively finer spatial subdivision. This process is analogous to reverse geocoding: translating a precise coordinate into a structured address composed of hierarchical components such as country, city, zipcode, street, and house number. Early tokens in our sequence correspond to broad regions, while later tokens refine the prediction to increasingly granular spatial detail. GeoToken then predicts this sequence of tokens autoregressively, predicting the next token for the next level

in the hierarchy conditioned on the image and all previously predicted levels. This sequential generation process, traversing the geographic hierarchy, allows the model to build its location estimate incrementally.

To guide this process and provide robust spatial context, GeoToken integrates retrieval-augmented context. Taking inspiration from [10], we first train dual image-gps and image-text encoders, using a CLIP-style contrastive loss [16]. This ensures that we get image embeddings that naturally align with embeddings of their corresponding GPS coordinates and location descriptions, providing strong priors for both retrieval and generation. Then, we use these encoders to retrieve context for our generation. Specifically, given an image input, we compute its embedding and use it to retrieve similar images from the training dataset. These retrieved images and their associated known token sequences serve as concrete “hints” to ground the generation process.

Furthermore, inspired by advancements in large language models, we leverage autoregressive decoding with test-time scaling techniques [17], [18]. This enables robust prediction and, crucially, inherently provides a natural way to manage uncertainty at each step of the prediction sequence. Utilizing techniques such as multi-answer sampling to explore alternative high-probability regions and maintain a set of plausible location hypotheses before committing to a final estimate allows us to extract robust predictions post-training. This mechanism of managing uncertainty mirrors human cognitive processes of considering alternatives and refining focus only when justified by strong evidence. A key benefit of this autoregressive, hierarchical approach and our training methodology is the ability to generate a rich pool of high-quality location candidates as the model explores the geographic hierarchy during decoding, offering a powerful alternative to the limited candidate sets from traditional methods. By framing the problem as a sequence generation over a geographic hierarchy, we move beyond fixed bins and retrieval limitations, offering a flexible and intuitive approach to worldwide geolocation

that mirrors human cognitive processes.

We validate the effectiveness of our approach through extensive experiments on the widely-used Im2GPS3k and YFCC4k benchmarks. To provide a comprehensive picture, we compare our performance in two distinct settings: one where our model operates independently (MLLM-free), and one where it is augmented by an MLLM. In the MLLM-free setting, our model surpasses other non-MLLM baselines on most metrics, achieving state-of-the-art performance with accuracy gains of up to 13.9%. When augmented with an MLLM, our model again outperforms all baselines, achieving state-of-the-art performance across all metrics. Importantly, the strong performance in the MLLM-free setting unlocks the ability to perform highly accurate geolocalization entirely on-device, ensuring user data remains secure—a critical advantage over API-dependent methods.

In summary, our contributions are:

- We introduce a novel hierarchical sequence prediction framework for worldwide image geolocalization, drawing inspiration from human reasoning and autoregressive language modeling.
- We propose a context-guided autoregressive decoding mechanism, integrating retrieval-augmented context to enhance robustness across diverse locations.
- Our autoregressive generation process inherently supports sampling an unlimited number of guesses, allowing navigation of different hierarchical paths in cases of uncertainty. We show that performance can be improved by test-time scaling methods, through generating a high-quality pool of samples and leveraging selection strategies to derive the final answer.
- Empirical evaluations show state-of-the-art performance on nearly all metrics in MLLM-free setting, establishing a new overall state-of-the-art when augmented with MLLM.

II. RELATED WORK

Our work builds upon advances in three primary areas: the long-standing computer vision task of image geolocalization, the paradigm of autoregressive generation popularized by large language models, and the framework of retrieval-augmented generation for grounding predictions in external knowledge.

A. Image Geolocalization

The task of estimating a photo’s geographic origin, known as image geolocalization, was pioneered by works like IM2GPS, which framed the problem as a large-scale image retrieval task [8]. Since then, approaches have generally fallen into three main categories: classification, retrieval, and hybrid methods.

a) Classification-based Approaches.: This paradigm partitions the Earth’s surface into a discrete grid and classifies an image into one of the cells. PlaNet formulates localisation as a multiclass problem over ~ 26000 cells [2]. Hierarchical refinements—splitting coarse cells only where training data

warrant—boost both resolution and sample efficiency (C-PlaNet [3]; ISNs [19]). Transformer backbones (TransLocator [4]) and vision-language pre-training (Clark et al. [5]) further improve accuracy. While effective at a coarse level, these methods are limited by their predefined grid resolution and struggle to make precise, continuous predictions [7].

b) Retrieval-based Approaches.: Concurrent to classification, retrieval-based methods match a query image against a large database of geotagged images, predicting the location of the best match or the average location of the top- k matches [8], [20]. These methods excel at recognizing specific landmarks but often fail in non-descript landscapes or regions with sparse data coverage in the retrieval gallery. Modern approaches have enhanced this paradigm with powerful deep learning features [20] and improved ranking strategies [6].

c) Hybrid and Modern Approaches.: Recently, the field has moved towards hybrid models that combine the strengths of both paradigms and leverage modern deep learning techniques. Contrastive learning has been used to learn joint embeddings for images and GPS coordinates, enabling both retrieval and direct regression-like prediction [7]. Other works have introduced sophisticated losses on semantic cells [11] or have explicitly used large multimodal models (MLLMs) to interpret visual cues and retrieved context, treating geolocalization as a generative task [9], [10]. Our work is inspired by this latter trend but focuses on a more fundamental sequence prediction approach that does not inherently depend on a pretrained MLLM for generation.

B. Autoregressive Generation

Autoregressive models are a fundamental class of generative models that produce complex, high-dimensional data one element at a time, where each new element is conditioned on all previously generated ones. This principle, which factorizes a joint distribution into a product of conditional probabilities, has long been the foundation of statistical language modeling [21].

The modern era of autoregressive generation was catalyzed by the introduction of the Transformer architecture [22], which replaced recurrent networks with a more parallelizable self-attention mechanism. This architectural shift, combined with massive datasets and computational scale, led to the development of Large Language Models (LLMs) like the GPT series [13], [14]. These models demonstrated an unprecedented ability to generate coherent and contextually relevant text, performing a wide range of tasks in a zero-shot or few-shot manner.

More recently, this paradigm has been extended to multimodal contexts. Multimodal Large Language Models (MLLMs) like Flamingo [23] and LLaVA [24] condition the autoregressive text generation process on visual inputs, enabling them to describe images, answer visual questions, and perform complex reasoning over visual data. Our work adapts this core autoregressive principle, but instead of generating natural language, we generate a sequence of hierarchical

geographic tokens conditioned on an image and its retrieved context.

C. Retrieval-Augmented Generation (RAG)

While large parametric models like LLMs store vast amounts of knowledge in their weights, they are prone to hallucination and their knowledge is static post-training. Retrieval-Augmented Generation (RAG) is a powerful framework designed to mitigate these issues by combining the parametric knowledge of a generator with a non-parametric external memory or database [25].

The standard RAG pipeline involves two stages. First, given a query, a retriever module searches a large corpus (e.g., Wikipedia) for relevant documents. Second, these retrieved documents are provided as additional context to a generator model (e.g., an LLM), which then produces the final output, grounded in the retrieved information. This approach has been shown to improve factuality, reduce hallucination, and allow for knowledge to be updated simply by changing the retrieval corpus, without costly retraining [25].

Initially developed for knowledge-intensive NLP tasks, the RAG concept has been extended to multimodal domains [26]. In computer vision, this often involves retrieving relevant images, text, or other data to provide context for a given visual task. Recent geolocalization models such as G3 [10] are prime examples of this trend, using retrieved images and their locations to inform a generative process. Our work builds directly on this paradigm, using a gallery of training images as our non-parametric memory and explicitly conditioning our autoregressive decoder on the retrieved context.

III. GEOTOKEN

An overview of GeoToken’s architecture and inference pipeline is illustrated in Figure 2. Given a query image, GeoToken first generates a location-aware embedding using a pretrained encoder. This embedding serves two purposes: it acts as a primary input to the prediction model and is used to retrieve visually similar images and their known locations from a training gallery. These retrieved locations are then tokenized into a hierarchical sequence. Finally, the embeddings of the query image and its retrieved neighbors, along with the tokenized neighbor locations, are fed into a transformer model [22] that autoregressively predicts the query’s location sequence. Optionally, various decoding and selection strategies can be employed to generate a pool of candidate locations and derive a final, robust prediction. The remainder of this section details each component of this framework.

A. Location Representation with Hierarchical S2 Tokens

As GeoToken predicts locations token-by-token in an autoregressive manner, we first require a method to convert geographic coordinates into a discrete sequence of tokens. To this end, we adopt Google’s S2 geometry for spatial indexing, which partitions the globe into a hierarchical quadtree structure.¹ At the coarsest level (level 0), the Earth is projected

onto a cube of six faces. Each cell is then recursively subdivided into four children (a quad-subdivision) to increase the resolution.

We represent a location as a sequence of L tokens derived from its S2 representation, covering levels 0 through $L - 1$. For our task, we use $L = 21$, which provides precision down to a few hundred meters. This process converts a location’s latitude and longitude into a token sequence:

$$T = [t_0, t_1, t_2, \dots, t_{L-1}],$$

where $t_0 \in \{0, \dots, 5\}$ is the S2 face token (level 0), and each subsequent token $t_i \in \{0, 1, 2, 3\}$ for $i \in \{1, \dots, L - 1\}$ encodes the quadrant at level i . This representation is inherently hierarchical; a shared prefix of length l signifies that two locations fall within the same S2 cell at level $l - 1$, providing an implicit notion of geographic proximity. For instance, locations within the same city will share a long common prefix, while those in different countries will diverge much earlier. This structure links token-space distance to real-world distance, as a small edit to a token sequence corresponds to moving to an adjacent region. The coarse-to-fine granularity also mirrors human-like descriptions, such as specifying a country, then a city, and finally a neighborhood.

B. Encoder Pretraining via Geo-Alignment

GeoToken relies on powerful embeddings to provide informative inputs to the predictive model and to retrieve relevant context for generation. To obtain these embeddings, we follow the approach proposed by prior work G3 [10] and train various encoders for encoding relevant information. This method learns expressive, location-aware representations by jointly aligning images with multi-modal geographical data: GPS coordinates and textual descriptions. Following G3, we define separate encoders for the image, GPS, and text modalities.

1) *Image Encoder*: For an input image I_i , a pretrained vision encoder $\mathcal{V}$ (e.g., ViT-L/14 [27] from CLIP [16]) first extracts raw visual features $e_i^{img_raw} = \mathcal{V}(I_i)$. These features are subsequently projected into two distinct embedding spaces using trainable feed-forward networks, $f_{text}^{img_proj}$ and $f_{gps}^{img_proj}$, to facilitate alignment with textual and GPS modalities, respectively:

- $e_i^{image_text} = f_{text}^{img_proj}(e_i^{img_raw})$: Image embedding for alignment with textual location descriptions.
- $e_i^{image_gps} = f_{gps}^{img_proj}(e_i^{img_raw})$: Image embedding for alignment with GPS coordinates.

For constructing a comprehensive image representation, embeddings are then concatenated to obtain final image embedding: $e_i^{image} = \text{concat}(e_i^{img_raw}, e_i^{image_text}, e_i^{image_gps})$.

2) *GPS Coordinate and Text Encoders*: Following G3, we encode raw GPS coordinates using an encoder, which applies a Mercator [28] projection followed by multi-scale Random Fourier Features (RFF) [29] and feed-forward networks. Similarly, textual descriptions (e.g., “Vestland, Norway”), obtained via reverse geocoding, are encoded using a pretrained text encoder and a trainable projection head.

¹<https://s2geometry.io/>

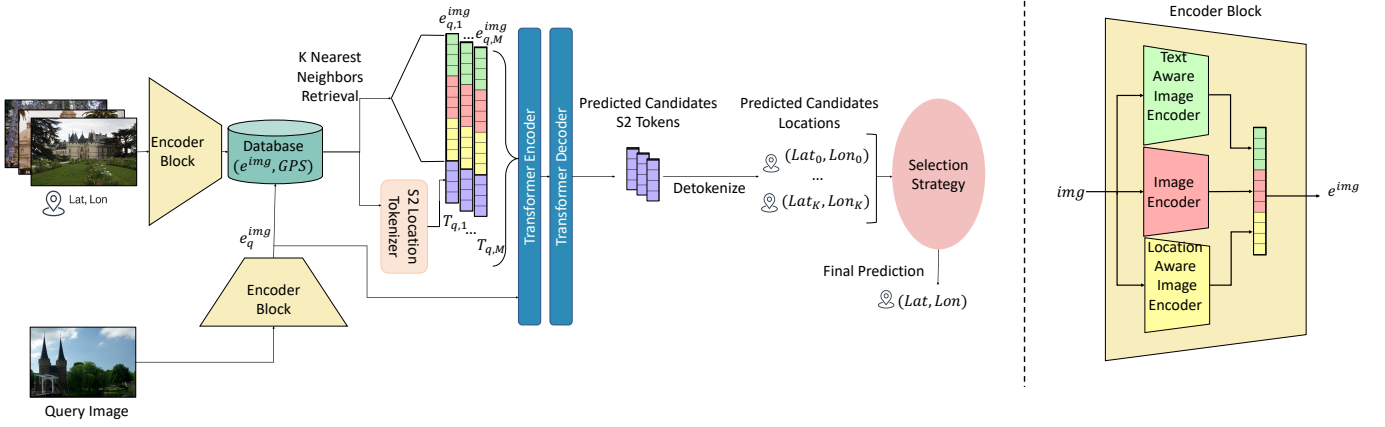


Fig. 2: The GeoToken pipeline for retrieval-augmented geolocation. A query image is encoded (1) and used to retrieve visually similar neighbors and their S2 location tokens from a gallery (2). This retrieved context grounds an encoder-decoder Transformer (3) that autoregressively predicts the final location as a hierarchical S2 token sequence. At test time, a pool of candidate locations is generated and a final prediction is chosen using a reranking strategy (4).

3) *Geo-Alignment Training Objective*: These encoders are trained jointly with a symmetric contrastive loss (InfoNCE) [30], which aligns the image embeddings with their corresponding GPS and text embeddings in a shared space. The total loss is:

$$\mathcal{L}_{\text{GeoAlign}} = \frac{1}{2}(\mathcal{L}_{\text{image, text}} + \mathcal{L}_{\text{image, gps}} + \mathcal{L}_{\text{text, image}} + \mathcal{L}_{\text{gps, image}})$$

This pretraining stage yields encoders that are highly attuned to location-indicative visual cues. For complete architectural details, we refer the reader to G3 [10].

C. Retrieval-Augmented Generation

With the encoders pretrained as described in Section III-B, the model is ready for its primary task: retrieval-augmented generation. This process uses an encoder-decoder transformer architecture. The encoder first processes the query image alongside its retrieved context to produce a contextualized memory representation. The decoder then attends to this memory to autoregressively generate the final location sequence, token by token.

a) *Context Retrieval*: The retrieval process begins with creating a gallery containing the image embedding e_i^{image} for every image i in the training dataset. As detailed in Section III-B, this embedding concatenates raw visual features with specialized projections aligned to GPS and text data, providing a powerful, multi-aspect representation for search.

Given a query image I_q , we compute its embedding e_q^{image} and use it to retrieve the top- M nearest neighbors from the gallery via cosine similarity. For each neighbor, we retrieve both its image embedding and its ground-truth S2 token sequence.

b) *Transformer Encoder Input*: The input to the transformer encoder is a sequence combining the query image with its retrieved context. Let e_q^{image} be the query image embedding. For each of the M nearest neighbors, let $e_{q,j}^{\text{image}}$ be its image embedding and $T_{q,j} = [T_{q,j}^0, \dots, T_{q,j}^{L-1}]$ be its S2 token sequence.

Each component is first projected into the transformer's hidden dimension d using dedicated embedding layers: an image embedding layer $E_{\text{img}}(\cdot)$ and an S2 token embedding layer $E_{\text{tok}}(\cdot)$. Let the projected vectors be $v_q = E_{\text{img}}(e_q^{\text{image}})$, $v_{q,j} = E_{\text{img}}(e_{q,j}^{\text{image}})$, and $\mathbf{t}_{q,j} = E_{\text{tok}}(T_{q,j})$. The full input sequence for the encoder, $\mathbf{X}$, is then formed by concatenation:

$$\mathbf{X} = \left[v_q \oplus \bigoplus_{j=1}^M (v_{q,j} \oplus \mathbf{t}_{q,j}) \right],$$

where $\oplus$ denotes concatenation along the sequence dimension. This yields a single sequence of length

$$1 + M \times (1 + L),$$

where each element is a vector in $\mathbb{R}^d$. Learned positional embeddings are added to differentiate the query from each neighbor's image and token block.

c) *Autoregressive Decoder*: The encoder processes the input sequence $\mathbf{X}$ via self-attention to produce a contextualized memory representation, $\text{Enc}(\mathbf{X})$. The causal transformer decoder then generates the S2 token sequence for the query, $(t_0, \dots, t_{L-1})$, one token at a time. At each step s , it models the conditional probability of the next token:

$$P(t_s | t_{<s}, \text{Enc}(\mathbf{X})).$$

To do this, the decoder attends to the full output of the encoder, leveraging both the query's visual context and the retrieved

geographic exemplars to inform its prediction for the next spatial token.

The decoder is effectively learning a language model over location tokens, conditioned on the rich context provided by the encoder. The autoregressive formulation models the joint probability of the sequence using the chain rule of probability:

$$P(t_1, \dots, t_L | \text{Enc}(\mathbf{X})) = \prod_{s=0}^{L-1} P(t_s | t_{<s}, \text{Enc}(\mathbf{X})).$$

This decomposition is well-suited for the task, as it allows the model to first focus on broad distinctions (the first few tokens) before gradually honing in on a precise location. If the input is ambiguous (e.g., between two neighboring cells at level 3), the model’s probability distribution at that step will reflect this uncertainty. The entire model is trained end-to-end with teacher forcing, using the a weighted sum of cross-entropy losses over all token positions as the objective.

D. Decoding Strategies

Generating a location from the trained model requires an autoregressive decoding process. The most straightforward approach, greedy decoding, selects the highest-probability token at each step. While fast, this deterministic method has two key weaknesses: it cannot represent uncertainty, forcing a single choice even when multiple regions are plausible, and it is susceptible to cascading errors, where an early mistake can derail the entire subsequent sequence.

To better handle ambiguity and improve robustness, we explore two families of decoding methods widely used in natural language processing: sampling with temperature and beam search.

1) *Sampling with Temperature*: Instead of deterministically picking the best token, we can sample from the probability distribution produced by the model at each step. This distribution is controlled by a temperature parameter T , which rescales the model’s output logits ℓ_t :

$$\mathbf{p}_t = \text{softmax}\left(\frac{\ell_t}{T}\right).$$

A temperature $T < 1$ sharpens the distribution, favoring high-probability tokens, while $T > 1$ flattens it, encouraging exploration. As $T \rightarrow 0$, sampling approaches greedy decoding. This controlled randomness is effective for exploring the solution space when the model is uncertain. By drawing multiple **independent** samples, we generate a set of plausible location sequences, increasing the probability that at least one candidate is highly accurate. Empirically, we find that a moderate temperature ($T \approx 0.5\text{--}0.7$) offers the best trade-off between reliability and diversity.

2) *Beam Search*: As a deterministic alternative, beam search maintains a “beam” of the top- B most probable partial hypotheses at each step. It systematically expands each partial sequence and retains the B new sequences with the highest cumulative log-probability:

$$\text{score}(\tau \oplus k) = \text{score}(\tau) + \log p(k | \tau),$$

where τ is a partial sequence and k is a candidate token. While beam search excels at finding high-probability sequences, its main drawback is its lack of stochasticity; an early, high-confidence error can trap the entire beam in an incorrect part of the search space.

E. Candidate Reranking and Selection Strategies

After generating a diverse pool of candidate sequences via decoding, a final step is required to derive the final prediction. We explore several strategies for this task.

1) *Log-Probability Selection*: The most direct method is to rely on the generative model’s own confidence scores. Each candidate sequence s has an associated cumulative log-probability:

$$\text{score}(s) = \sum_{j=0}^{L-1} \log p(k_j | k_1, \dots, k_{j-1}).$$

We select the candidate with the highest log-probability, which favors the sequence that the model itself considers most likely.

2) *Reward Model Reranking*: Another approach is to train a separate reward model to predict the accuracy of a given candidate sequence. To do this, we take a rather simple approach and discretize the continuous prediction error into binary bins corresponding to the evaluation distance thresholds <200 km and >200 km. The process is as follows:

- 1) Generate Dataset: For each image in the training set, sample a set of candidate location sequences using our trained model.
- 2) Label Data: For each candidate, decode its coordinates and compute its haversine distance to the ground truth, then label it with the corresponding error bin.
- 3) Train Scorer: Train a classifier $c_\psi(s_i)$ on this dataset to predict the correct error bin for any given sequence s_i .
- 4) Select Best: At inference time, apply the trained scorer to all candidates in the pool and select the one with the highest predicted probability of being in the smallest-error bin ($b = 0$):

$$s^* = \arg \max_i [c_\psi(s_i)]_{b=0}.$$

3) *Similarity-based Selection*: This strategy leverages the shared embedding space learned during our geo-alignment pretraining (Section III-B). For each candidate location sequence, we first decode it to get its GPS coordinate and then generate its corresponding location embedding using our pretrained GPS encoder. We then select the candidate whose location embedding exhibits the highest cosine similarity with the query image’s embedding.

4) *MLLM-as-a-Judge*: This strategy employs a large multimodal model (MLLM) to arbitrate among the generated candidates. The MLLM is provided with the query image and the pool of candidates and can be used in one of two modes:

- **Pool-Selection Mode**: The MLLM is prompted to choose the best option from a list of candidate coordinates.

- **Free-Generation Mode:** The MLLM is allowed to either pick one of the provided candidates or generate an entirely new coordinate if it determines none are sufficiently accurate.

The final prediction is then parsed from the MLLM’s textual response.

IV. EXPERIMENTS

In this section, we detail a comprehensive evaluation designed to validate our hierarchical sequence prediction approach. We benchmark GeoToken against a wide array of state-of-the-art methods on two standard geolocalization datasets: IM2GPS3K and YFCC4K.

A. Experimental Setup

1) *Datasets:* Our experiments leverage one training corpus and two distinct evaluation benchmarks to assess both in-distribution and out-of-domain generalization.

- **MP16-Pro (Training):** This is our large-scale training corpus, derived from the original MP16 dataset. It contains approximately 4.1 million Flickr images. Following the procedure in [10], each image is annotated with multi-level geographic text (e.g., city, country, continent) using Nominatim. This dataset is used for both the initial CLIP-style geo-alignment and for training the main GeoToken model.
- **IM2GPS3K (Evaluation):** This benchmark contains 3,000 diverse, globally-distributed images and is a strong test of out-of-domain generalization [8]. Its emphasis on rural and non-landmark scenes makes it particularly challenging for retrieval-based methods.
- **YFCC4K (Evaluation):** This benchmark is a 4,000-image subset of the YFCC100M dataset [31]. In contrast to IM2GPS3K, its distribution of urban scenes and popular landmarks more closely mirrors our training data, testing the model’s performance on more familiar-looking scenes.

B. Baselines

We compare GeoToken with the most widely-used models in image geolocalisation:

- **k -NN ($\sigma=4$)** [6]. Returns the mean location of the k nearest visual neighbours; shrinking k lowers the Gaussian bandwidth and approaches plain 1-NN.
- **PlaNet** [3]. Casts the task as a single multi-class classification problem by partitioning the globe into thousands of cells.
- **C-PlaNet** [3]. Improves PlaNet by letting overlapping coarse cells vote for finer-grained intersections.
- **ISNs** [19]. Adds a parallel scene-context branch (indoor, urban, natural) and fuses it with hierarchical cell scores.
- **TransLocator** [32]. Processes the raw image and its semantic-segmentation map through a dual-stream transformer.

- **GeoDecoder** [33]. Applies cross-attention between coarse and fine tokens to reduce error propagation in deep hierarchies.
- **GeoCLIP** [7]. Learns a GPS encoder that aligns CLIP image embeddings with location vectors.
- **Img2Loc** [34]. Treats geolocalisation as retrieval-augmented generation: retrieved coordinates become tokens in an MLLM prompt.
- **PIGEON / PIGEOTTO** [11]. Creates semantic geo-cells and introduces a distance-aware smoothing loss that softens class boundaries.
- **G3** [10]. Combines large-scale retrieval with a generative prior, drawing several candidate coordinates before a final selection step using the similarity-based approach.

C. Implementation Details

a) *Hierarchical S2 Tokenization:* Every latitude-longitude coordinate is converted into a 21-token sequence using Google’s S2 geometry library at level 20. This sequence consists of one token for the initial cube face (from a vocabulary of 6) and 20 subsequent quad-tree tokens (each from a vocabulary of 4) that progressively refine the location. A single embedding table is shared across all 21 positions.

b) *Model Architecture:* GeoToken is a 10-layer encoder-decoder Transformer ($d_{\text{model}}=512$, 8 attention heads, 1024-dim FFN). It processes a concatenated sequence of specialized embeddings derived from a frozen CLIP ViT-L/14 backbone. The input consists of a learnable $[\text{CLS}]$ token followed by projections representing the query image, its ground-truth location and text metadata, and context from its top-15 retrieved neighbors from the MP16-Pro gallery.

c) *Training:* Training proceeds in two stages. First, we perform Geo-Alignment by training the image and location encoders with a symmetric InfoNCE loss for 10 epochs to align their embeddings. Second, the full GeoToken model is trained for 50 epochs on MP16-Pro using AdamW (initial LR 5×10^{-5} , weight decay 10^{-6}). Batches of 2048 are trained on a single NVIDIA GH200 GPU. The loss function is a position-weighted cross-entropy (CE) that penalizes errors at coarser levels of the S2 hierarchy more heavily:

$$\mathcal{L} = \frac{1}{\sum_t w_t} \sum_{t=0}^{20} w_t \text{CE}(\hat{y}_t, y_t), \quad \text{where } w_t = 2.0 - \frac{t}{20}$$

d) *Evaluation Protocol:* We report accuracy at standard distance thresholds ($\{1, 25, 200, 750, 2500\}$ km) and median geodesic error. Our experiments use three general evaluation protocols to assess different aspects of our framework:

- 1) **Single Deterministic Prediction:** Evaluates a single output from the model, produced by a deterministic decoding strategy like greedy decoding or beam search. This protocol is used to assess the core model’s performance against MLLM-free baselines.
- 2) **Selected-from-Pool Prediction:** Evaluates the final prediction after a selection strategy is applied to a pool of generated candidates (typically $K = 30$ candidates from

TABLE I: Overall localization accuracy (%) on IM2GPS3K and YFCC4K, with median error (km).

Method	IM2GPS3K					YFCC4K				
	1 km	25 km	200 km	750 km	2500 km	1 km	25 km	200 km	750 km	2500 km
[L]kNN	7.2	19.4	26.9	38.9	55.9	2.3	5.7	11.0	23.5	42.0
PlaNet	8.5	24.8	34.3	48.4	64.6	5.6	14.3	22.2	36.4	55.8
C-PlaNet	10.2	26.5	34.6	48.6	64.6	7.9	14.8	21.9	36.4	55.5
ISN	10.5	28.0	36.6	49.7	66.0	6.5	16.2	23.8	37.4	55.0
TransLocator	11.8	31.1	46.7	58.9	80.1	8.4	18.6	27.0	41.1	60.4
Clark et al.	12.8	33.5	45.9	61.0	76.1	10.3	24.4	33.9	50.0	68.7
GeoCLIP	14.1	34.5	50.7	69.7	83.8	9.6	19.3	32.6	55.0	74.7
PIGEON	11.3	36.7	53.8	72.4	85.3	10.4	23.7	40.6	62.2	77.7
GeoToken	16.8	39.6	53.8	70.8	85.0	24.3	35.3	46.6	64.2	78.6

TABLE II: Comparison of Localization accuracy (%) using GeoToken, Img2Loc, and G3 under the MLLM-assisted setting using Gemini-2.0-Flash on IM2GPS3K and YFCC4K.

Method	IM2GPS3K					YFCC4K				
	1 km	25 km	200 km	750 km	2500 km	1 km	25 km	200 km	750 km	2500 km
Img2Loc	16.4	42.5	55.6	72.2	85.3	18.7	31.6	43.8	62.0	76.1
G3	17.2	44.4	59.1	74.6	86.8	22.9	37.2	50.3	66.9	79.9
GeoToken (Pool-Selection Mode)	18.8	45.0	59.3	75.2	87.7	24.7	37.7	50.3	67.0	80.5
GeoToken (Free-Generation Mode)	19.0	46.0	60.1	76.6	88.8	25.4	38.5	51.4	68.0	81.0

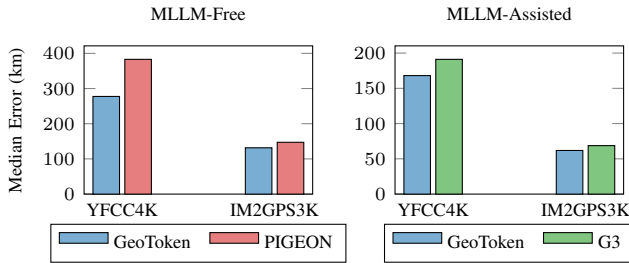


Fig. 3: Comparison of the median localization error (km) on YFCC4K and IM2GPS3K of GeoToken and prior state-of-the-art approaches. **Left:** MLLM-Free (GeoToken vs. PIGEON). **Right:** MLLM-Assisted (GeoToken vs. G3).

temperature sampling). The specific selection strategy varies by experiment and includes our main MLLM-as-a-Judge pipeline as well as other methods analyzed in our ablations (e.g., log-probability, similarity-based).

- 3) Candidate Pool Quality (Ideal Selector): Measures the upper-bound potential of our generative model by reporting the accuracy of the best possible candidate within the generated pool (i.e., the closest-in-pool). This is used in our ablation studies to analyze the quality of the candidate set itself, independent of the selection strategy.

D. Main Quantitative Results

Baseline methods fall into two categories: (1) those that work locally and do not rely on a powerful MLLM for prediction (MLLM-free), and (2) those that leverage MLLMs (MLLM-assisted). To ensure fair comparisons and to demonstrate the performance of GeoToken as a standalone model, we separate our evaluations into these two settings. In the MLLM-free setting, we evaluate a single greedy decoding prediction from GeoToken against comparable baselines. In the MLLM-assisted setting, we compare our full pipeline (sampling 30 candidates with an MLLM-as-a-Judge) against

G3 and Img2Loc. For a direct comparison, we reproduce the results for these baselines using the same MLLM judge (Gemini 2 Flash) as our own method. Our reproduced results for these baselines are generally higher than those originally reported, confirming a fair and strong comparison.

As shown in Table I, in the MLLM-free setting, GeoToken’s greedy prediction establishes a new state of the art on nearly all metrics across both benchmark datasets. On the challenging IM2GPS3K dataset, GeoToken significantly improves accuracy at finer scales (e.g., 1 km and 25 km), where prior methods often struggle. The improvement is even more pronounced on YFCC4K, where at 1 km GeoToken more than doubles the accuracy of the next-best method and maintains its lead across all subsequent radii.

In the LLM-assisted setting (Table II), GeoToken’s performance is further amplified. It consistently outperforms other MLLM-augmented methods like Img2Loc and G3 across all distance thresholds on both datasets. These results underscore the strength of our hierarchical generation approach, which provides a superior candidate pool for the MLLM judge to refine. As shown in Figure 3, these accuracy gains are reflected in a lower median prediction error compared to prior art in both evaluation settings.

E. Ablation Studies

1) *Effect of Sample Pool Size and Temperature:* To understand how the quality of the candidate pool is affected by the number of candidates (k) and the sampling temperature (T), we perform a “closest-in-pool” ablation on both YFCC4K and IM2GPS3K. We generate 30 candidate sequences per image and vary:

- $k \in \{5, 10, 15, 20, 30\}$, the number of sampled candidates per image.
- $T \in \{0.2, 0.5, 0.7, 1.2\}$, the sampling temperature.

Figure 4 illustrates the median error of the best guess in the candidate pool as a function of k for each temperature.

TABLE III: Ablation of candidate selection strategies (%) on IM2GPS3K and YFCC4K.

Method	IM2GPS3K					YFCC4K				
	1 km	25 km	200 km	750 km	2500 km	1 km	25 km	200 km	750 km	2500 km
Ideal Selector	33.1	59.2	77.3	90.1	95.7	39.1	56.3	75.2	89.9	96.3
Log-Probability	16.4	38.6	52.1	68.6	83.2	26.1	36.6	47.2	63.9	78.5
Beam Search (beam=2)	16.9	39.7	53.3	69.9	84.2	25.8	36.5	47.4	64.5	78.4
Beam Search (beam=3)	16.2	38.7	52.7	69.2	83.7	26.2	36.4	47.4	64.2	78.7
Beam Search (beam=4)	15.7	38.0	51.9	69.0	83.2	26.5	36.7	47.5	64.3	78.6
Reward Model (bin-classifier)	14.5	35.1	48.0	65.5	79.9	19.0	29.5	42.2	60.0	74.9
CLIP Similarity	14.1	36.7	51.7	69.9	83.7	19.0	30.2	43.5	61.4	77.0
MLLM-as-a-Judge (Pool-Selection)	18.8	45.0	59.3	75.2	87.7	24.7	37.7	50.3	67.0	80.5
MLLM-as-a-Judge (Free-Generation)	19.0	46.0	60.1	76.6	88.8	25.4	38.5	51.4	68.0	81.0

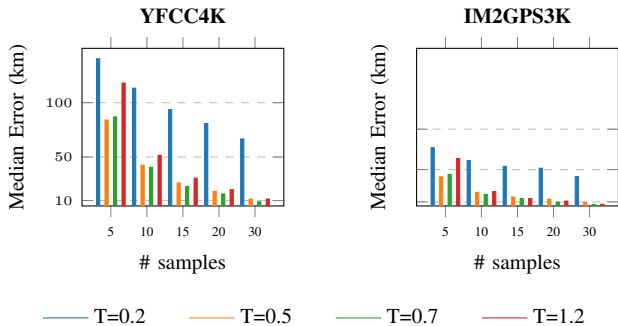


Fig. 4: Median error of best closest-in-pool using different numbers of samples and temperatures on both datasets.

Key observations include: (1) As expected, the median error considerably decreases as k increases, showing the benefit of a larger pool. (2) Intermediate temperatures ($T = 0.5$ or $T = 0.7$) perform best, balancing exploration and exploitation. (3) YFCC4K exhibits larger median errors than IM2GPS3K across all settings due to higher scene diversity. Based on these results, we adopted $T = 0.7$ and $k = 30$ as default sampling hyperparameters.

2) *Effect of Candidate Selection Strategy*: In our main results, we use MLLM-as-a-Judge as our default selection strategy. Here, we compare it against beam search and other methods detailed in Section III-E. For each query, we generate 30 candidates ($T = 0.7$) and apply different selection strategies. Table III shows the results. Ranking by log-probability and beam search provide strong performance, outperforming greedy prediction on many metrics. In contrast, CLIP similarity and the Reward Model perform poorly. The MLLM-as-a-Judge approaches offer a significant boost, with free-generation being the most effective. However, a significant gap remains between our best selector and the “Ideal Selector” (which would always pick closest-in-pool). This indicates that the candidate pool contains significantly higher-quality samples than our current selection strategies can identify, suggesting great potential for future work on more powerful selection mechanisms.

V. DISCUSSION: PRIVATE INFERENCE

Unlike methods that rely heavily on MLLM APIs for their core prediction, GeoToken’s architecture is able to provide

strong performance in its MLLM-free configuration. This has critical privacy implications. Because all inference can be performed locally without sending data to a third party, making GeoToken suitable for on-device or private-server applications. Users can geolocate their images, which may contain sensitive personal information, without exposing them to the risks associated with external cloud services. This “local-first” capability ensures users retain full control over their data, a significant advantage over other leading models.

VI. CONCLUSION

Inspired by advances in large language modeling and hierarchical decoding, we have presented GeoToken, a hierarchical sequence-prediction framework that mirrors human coarse-to-fine reasoning for worldwide image geolocalization. By treating location as a sequence of S2-cell tokens, our model first narrows down broad regions and then refines its prediction step by step, allowing it to capture both high-level and fine-grained geographic cues. This autoregressive setup not only achieves state-of-the-art accuracy without any external LLM, outperforming all non-MLLM baselines by large margins at nearly every distance threshold, but also naturally supports sampling multiple plausible location hypotheses at inference time.

Sampling from GeoToken provides two key benefits. First, generating a pool of candidate coordinates lets the model explicitly manage uncertainty: if the visual evidence is ambiguous, multiple hierarchical paths can be explored before committing. Second, this sampling process is entirely local, preserving user privacy by avoiding reliance on external retrieval services or cloud-based APIs. When an MLLM judge is added to select or refine among these candidates, GeoToken further extends its lead, even over other LLM-augmented pipelines, demonstrating that our hierarchical predictions remain superior whether or not an LLM is used.

Looking forward, GeoToken’s flexibility invites easy integration of additional modalities (e.g., timestamps, low-resolution satellite imagery) or more advanced retrieval schemes, further enhancing robustness in underrepresented regions. By releasing our code, pretrained weights, and the MP16-Pro splits, we hope to encourage future work that leverages hierarchical token prediction and uncertainty-aware sampling to push the frontiers of worldwide image geolocalization.

VII. ACKNOWLEDGMENTS

This research has been funded in part by NSF grants IIS-2128661 and 1956435, and NIH grant 5R01LM014026. Opinions, findings, conclusions, or recommendations expressed are those of the author(s) and do not necessarily reflect the views of any sponsors, such as NSF or NIH.

REFERENCES

- [1] M. Larson, M. Soleymani, G. Gravier, B. Ionescu, and G. J. Jones, “The benchmarking initiative for multimedia evaluation: Mediaeval 2016,” *IEEE MultiMedia*, vol. 24, no. 1, pp. 93–96, 2017.
- [2] T. Weyand, I. Kostrikov, and J. Philbin, “Planet - photo geolocation with convolutional neural networks,” in *ECCV*, 2016, pp. 37–55.
- [3] P. H. Seo, T. Weyand, J. Sim, and B. Han, “Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 536–551.
- [4] S. Pramanick, E. M. Nowara, J. Gleason, C. D. Castillo, and R. Chel-lappa, “Where in the world is this image? transformer-based geo-localization in the wild,” *arXiv preprint arXiv:2204.13861*, 2022.
- [5] B. Clark, A. Kerrigan, P. P. Kulkarni, V. V. Cepeda, and M. Shah, “Where we are and what we’re looking at: Query based worldwide image geo-localization using hierarchies and scenes,” in *CVPR*, 2023.
- [6] N. Vo, N. Jacobs, and J. Hays, “Revisiting im2gps in the deep learning era,” in *ICCV*, 2017, pp. 2621–2630.
- [7] V. V. Cepeda, G. K. Nayak, and M. Shah, “Geoclip: Clip-inspired alignment between locations and images for effective worldwide geo-localization,” *NeurIPS*, 2023.
- [8] J. Hays and A. A. Efros, “Im2gps: estimating a photograph’s location using internet-scale imagery,” in *CVPR*, 2008, pp. 1–8.
- [9] Z. Zhou, N. Lao, J. Zhang, and L. Mu, “Img2loc: Revisiting image geolocalization using multi-modality foundation models and image-based retrieval-augmented generation,” *arXiv preprint arXiv:2309.17421*, 2023.
- [10] P. Jia, Y. Liu, X. Li, Y. Wang, Y. Du, X. Han, X. Wei, S. Wang, D. Yin, and X. Zhao, “G3: An effective and adaptive framework for worldwide geolocalization using large multi-modality models,” *NeurIPS*, 2024.
- [11] L. Haas, M. Skreta, S. Alberti, and C. Finn, “Pigeon: Predicting image geolocations,” in *CVPR*, 2024.
- [12] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” *OpenAI blog*, 2018.
- [13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [15] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [16] A. Radford, J. W. Kim, T. Xu, G. Brockman, I. Sutskever, and S. van der Burgh, “Learning transferable visual models from natural language supervision,” *arXiv preprint arXiv:2103.00020*, 2021.
- [17] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, “Self-consistency improves chain of thought reasoning in language models,” *arXiv preprint arXiv:2203.11171*, 2022.
- [18] C. Snell, J. Lee, K. Xu, and A. Kumar, “Scaling llm test-time compute optimally can be more effective than scaling model parameters,” *arXiv preprint arXiv:2408.03314*, 2024.
- [19] E. Muller-Budack, K. Pustu-Iren, and R. Ewerth, “Geolocation estimation of photos using a hierarchical model and scene classification,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 563–579.
- [20] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.
- [21] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.
- [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [23] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, “Flamingo: a visual language model for few-shot learning,” *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.
- [24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [25] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.
- [26] M. Yasunaga, A. Aghajanyan, W. Shi, R. James, J. Leskovec, P. Liang, M. Lewis, L. Zettlemoyer, and W.-t. Yih, “Retrieval-augmented multi-modal language modeling,” *arXiv preprint arXiv:2211.12561*, 2022.
- [27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [28] J. P. Snyder, *Map Projections—A Working Manual*, ser. US Geological Survey Professional Paper 1395. US Government Printing Office, 1987.
- [29] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” *Advances in neural information processing systems*, vol. 33, pp. 7537–7547, 2020.
- [30] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [31] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research,” *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.
- [32] S. Pramanick, E. M. Nowara, J. Gleason, C. D. Castillo, and R. Chel-lappa, “Where in the world is this image? transformer-based geo-localization in the wild,” in *European Conference on Computer Vision*. Springer, 2022, pp. 196–215.
- [33] B. Clark, A. Kerrigan, P. P. Kulkarni, V. V. Cepeda, and M. Shah, “Where we are and what we’re looking at: Query based worldwide image geo-localization using hierarchies and scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 182–23 190.
- [34] Z. Zhou, J. Zhang, Z. Guan, M. Hu, N. Lao, L. Mu, S. Li, and G. Mai, “Img2loc: Revisiting image geolocalization using multi-modality foundation models and image-based retrieval-augmented generation,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2749–2754.