GeorAIn: Demystifying GeoGuessr

Fayez Navid Anwar
Stanford University
Stanford, CA
fayez@stanford.edu

Abstract

Vision-based geolocation is used to deduce the location an image was taken at, with applications in social media (auto-tagging), forensics, and disaster response. Using the game GeoGuessr as a motivator, our work aims to understand the key image features that can be used to deduce their locations. Using Mapillary's Street-level Sequences (MSLS) corpus [3], comprising 1.6 million crowdsourced dash-cam images across 30 cities, our work demonstrates that a straightforward single-image, single-stage regression model achieves competitive localization accuracy, with sub-500 km average error, without extensive computational resources. Training on 7% of MSLS (113,000 frames), our model achieves a meean error of 489 km and median error of 102km, successfully localizing 49.6% of images within a 100km radius, beating a naive k-nearest neighbors baseline by 7x. We utilize visualizations with Grad-CAM to reveal that the model implicitly identifies key visual cues, such as lane-marking colors, architectural features, vegetation shapes, and sky coloration — all of which are commonly employed by top GeoGuessr players. Our findings suggest that effective geolocation can be achieved without sequence-based context or retrieval-based methods when leveraging robust pretrained vision models, with sufficient generalizability.

1. Introduction

GeoGuessr is an online geography game where you are shown a street view image from a random location in the world, and you have to pinpoint where that is on the map. The closer your pin is to the true location, the higher you score. The top human players are consistently able to guess within the correct city ($\approx 25km$) or locale ($\approx 100km$) with remarkable speed, combining visual cues such as the color of the dirt, the colors and shapes of road markings, the relative position of the sun in the sky, the number of electric poles lining roads, and so on. As part of this project, we've been investigating the possibility of training neural networks to play GeoGuessr, providing a single street view image as input to the network, which then outputs its best guess at the coordinates for the image.

Formally, the problem is: given a single RGB street-level image, predict the location at which the image was taken, in the form of (lat, long) coordinates or an equivalent 3D unit vector. This task is made difficult by two key factors: images may be taken on streets where there are no well-known landmarks in sight, and the same scenery may be different depending on the time of day as well as the season and weather conditions.

We employ the Mapillary Street-level Sequences (MSLS) Dataset to train a network for this, subsampling approximately 113,000 images ($\tilde{5}$,000 per city) to enable comprehensive cross-city generalization. Using the raw RGB images, we leverage a pretrained CLIP ViT-B/16 model as a fixed feature extractor to generate compact visual embeddings, coupledd with a two-layer multilayer perceptron trained using the von Mises–Fisher loss (vMF loss) function, to output a 3-D unit vector $\hat{\mathbf{p}} = [x, y, z]^\top \in \mathbb{S}^2$ that points from the center of the Earth to the predicted location on the surface.

2. Related Work

Before embarking on this project, we also spent considerable time understanding the work that has preceded this investigation, for image-based geolocation problems. Some key papers, and the findings from them that proved most helpful, are summarized below:

- **PlaNet** [9] first cast global geolocation as a classification task over S2 cells; their coarse-cell loss motivates our choice of city cell splitting and provides a reference for accuracy at various levels: continent-level, country-level, city-level, and street-level.
- **Im2GPS** [5] introduced KNN retrieval for outdoor photos. It uses a database of 6.5M Flickr images, and finds nearest neighbors based on six global image descriptors. This approach forms the conceptual basis of our non-parametric baseline.

Later work on Im2GPS such as **Vo *et al.*** [8], showed that features learned for cell classification can also be used as embeddings for retrieval. This offered high fine-scale accuracy while requiring far fewer reference images than PlaNet. Retrieval can localize quite a bit, even getting the exact street correct if a very close scene exists in the gallery. However, there continue to be practical hurdles, given the search cost and amount of coverage / gallery size needed at query time.

- **PIGEON** [4] proposes a two-stage, planet-scale system that first pretrains a ViT backbone with a multi-task contrastive objective and semantic geo-cell labels, followed by refining coarse predictions with a retrieval module. The authors were able to achieve SOTA results on GeoGuessr-style-tasks.
- **Pure Coordinate Regression** is another approach that seems attractive for its apparent simplicity. However, it is notoriously unstable at global scale in practical settings. We came across several past works that established this, including, for example, **De Brébisson *et al.*** [2] who reported poor convergence on it when training neural networks to predict taxi destinations. Similarly, the recent **OpenStreetView-5M** benchmark [1] confirmed that a CLIP-ViT + MLP regressor trails classification-augmented baselines.

  Similar to PlaNet, the winning strategy now seems to be coarse-to-fine: predict a rough region first, then regress or retrieve within it, so that we can use some amount of visual and geographical reasoning to narrow down the search scope, followed by a memory component to get more precise.

3. Methods

We cast world-scale geolocation as *metric regression on the unit sphere*. For clarity, we first formalise the task (Sec. 3.1), then describe our network architecture (Sec. 3.2), the von Mises–Fisher training objective (Sec. 3.3), the optimisation schedule (Sec. 3.4), and finally the evaluation protocol and baselines (Sec. 3.5). A basic schematic pipeline is shown in Fig. 1.

3.1. Problem formulation

Let $I \in \mathbb{R}^{H\times W\times 3}$ be a street-level RGB image captured somewhere on Earth. Its ground–truth position is represented by a 3-D unit vector $\mathbf{p} = \big[\cos\phi\cos\lambda,\ \ \cos\phi\sin\lambda,\ \ \sin\phi\big]^\top \in \mathbb{S}^2$, where ϕ and λ denote latitude and longitude, respectively. Our goal is to learn a function $f_\theta\colon \mathbb{R}^{H\times W\times 3} \to \mathbb{S}^2$, parameterised by θ, that predicts $\hat{\mathbf{p}} = f_\theta(I)$ so as to minimise the *great-circle distance*

$$d_{\mathrm{gc}}\big(\mathbf{p}, \hat{\mathbf{p}}\big) := R_\oplus\ \arccos\!\big(\mathbf{p}^\top \hat{\mathbf{p}}\big), \tag{1}$$

with $R_\oplus = 6\,371$ km the Earth's mean radius.

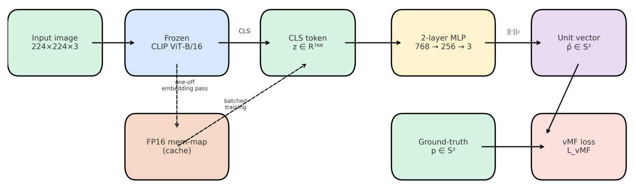

Figure 1: Model Pipeline

3.2. Architecture overview

Our network has two key components (Fig. 1):

(1) Frozen CLIP backbone. We adopt **CLIP ViT-B/16** [7], a 12-layer Vision Transformer with 768-D hidden size and 16-pixel patches. Given a 224×224 crop I', the model outputs a sequence of $1 + 14^2$ tokens. We keep the ImageNet-scale parameters *frozen* and retain only the `[CLS]` token $\mathbf{z} \in \mathbb{R}^{768}$ as a global visual descriptor. Freezing yields two key advantages: *(i)* smaller memory footprint (no activations stored for backward), *(ii)* training stability on a 100-k-image dataset, without having to retrain lower-level features such as edge-detection etc.

(2) vMF regression head. A two-layer MLP projects the feature to $\mathbb{R}^3$:

$$\mathbf{h} = W_2\,\sigma\big(W_1\,\mathbf{z}\big), \qquad W_1 \in \mathbb{R}^{256\times 768},\ W_2 \in \mathbb{R}^{3\times 256}.$$

We obtain the predicted unit vector by $\hat{\mathbf{p}} = \mathbf{h}/\|\mathbf{h}\|_2$. The head has only $260\,$k parameters and can train relatively quickly, taking less than one hour on an A100 GPU.

3.3. von Mises Fisher loss

To model directional uncertainty we treat predictions as the mean direction of a 3-D von Mises–Fisher (vMF) distribution [6]:

$$p\big(\mathbf{p} \mid \hat{\mathbf{p}}, \kappa\big) = C_3(\kappa)\ \exp\!\big(\kappa\,\hat{\mathbf{p}}^\top\mathbf{p}\big), \qquad C_3(\kappa) = \frac{\kappa}{2\pi\big(e^{\kappa} - e^{-\kappa}\big)}.$$

Here κ is the concentration parameter ($\kappa = 0$ is uniform). We optimize the negative log likelihood

$$\mathcal{L}_{\mathrm{vMF}} = -\kappa\,\hat{\mathbf{p}}^\top\mathbf{p} + \log C_3(\kappa). \tag{2}$$

Directly learning κ is unstable for $\kappa > 50$, due to numerical instability as the normalizer requires evaluating e^{kappa}, which overflows 32-bit and 64-bit floating point at $\kappa \approx 88$, so we instead learn $\tilde{\kappa}$ and set $\kappa = \mathrm{softplus}(\tilde{\kappa}) + 10^{-3}$. For $\kappa > 50$ we use the large-κ approximation $\log C_3(\kappa) \approx \log\kappa - \kappa - \log(2\pi)$, avoiding numerical overflow.

Phase	Epochs	Frozen	LR	Batch
Embed	–	all	–	128
Probe	40	CLIP	2.5×10^{-3}	2 048
Fine-tune	2	last 6 unf.	5×10^{-5}	64

Table 1: Training schedule on a single Colab A100.

3.4. Training strategy

Embedding pass. To manage OOM crashes, we streamed all training images once through the frozen backbone, while storing CLS tokens are stored as a mem-mapped array.

Linear probe. We train the head + $\tilde{\kappa}$ for 40 epochs using AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$) and a cosine decay from 10^{-2} to 10^{-4} (Tab. 1).

Light fine-tuning. Unfreezing the last six transformer blocks adds 11 M parameters. We train for two epochs at LR $= 5 \times 10^{-5}$ with mixed precision, gradient checkpointing, and early-stopping on convergence.

Data augmentation. Given the need to have our model generalize to a variety of real-world variations in conditions, such as weather, time of day, and seasons, we augmented the input images in ways that would perturb relevant aspects, but not result in a misrepresentation of useful information: we enabled random cropping (0.8 - 1.0), color jitter, and gaussian blur, but we disabled horizontal flips so that cues such as left/right traffic signage and the position of the sun would be maintained as is.

3.5. Evaluation protocol and baselines

Splits. We use 23 cities, with their datasets split between train (90%), validation (5%), and test (5%), by Sequence ID. Within a city we split by *sequence_key* so no contiguous frames leak across splits.

Metrics. Primary: mean and median great-circle error (km). Auxiliary: Inspired by GeoGuessr scoring tiers, we measure hit-rates within 25km and 100km.

Baseline. *Random guess:* As a naive baseline, we sample a location uniformly on Earth's surface, i.e. lat $\tilde{\phi} \sim U[-\frac{\pi}{2}, \frac{\pi}{2}]$ and long $\tilde{\lambda} \sim U[-\pi, \pi]$. The expected great-circle error for such a guess is $\mathbb{E}[d_{\text{gc}}] = R_{\oplus}\,\pi/2 \approx 10\,008\,\text{km}$, as this is approx. a quarter of Earth's circumference. The median error we'd expect for this would be $d_{50} = R_{\oplus}\,\arccos(1/2) \approx 6\,672\,\text{km}$.

For our aux. metrics, the probability of falling within a distance d of the ground truth is $P(d) = 1 - \cos(d/R_{\oplus})$; thus within 25km we have $P(25) = 7.7 \times 10^{-6}$ (0.0008%) and within 100km, $P(100) = 1.2 \times 10^{-4}$ (0.012%).

This provides an early lower bound for our network, as a sanity check.

4. Dataset and Features

A key component of past studies in the subject, such as PlaNet [9] and PIEGON [4], has been the use of licensed Google Street View data, which is prohibitively expensive to acquire at the scale required for Deep Learning applications independently in an academic setting.

To overcome these limitations, we have instead relied on the Mapillary Street-level Sequences (MSLS) dataset. MSLS contains 1.6 million Creative Commons-licensed dash-cam images, featuring 30 major cities across six continents. The images have a high degree of variability due to natural factors, such as seasons and day-to-day weather conditions, which make it valuable for model training, while being free to access.

We subsampled MSLS to help train our model efficiently, sampling 113,000 images from 23 cities with sufficient data, with 5000 images per city.

MSLS data is in the form of sequences of images, with (lat, long) values for each image, and a sequence ID. Given the data is from dashcams, images from the same sequence are often temporally close. Thus, we decided to split the training, validation, and test datasets by sequence IDs, so that each sequence would only be used in *one* of the splits.

Figure 2: Amsterdam

key	value
idx	6825
key	-Mi4ZXIOjWS...
lon	4.83851
lat	52.35784
ca	97.94671
captured_at	2018-11-03
pano	False
sequence_key	11667
frame_number	sgxkml...

Table 2: Metadata for Amsterdam Image (Figure 2).

5. Experiments, Results & Discussion

In this section, we describe the experimental setup (datasets, training approach, and hyperparameter search), and define the evaluation metrics used for our project. After this, we present our results, discussing and analyzing them with quantitative and qualitative techniques.

5.1. Experimental Setup

Dataset split. We use the *Mapillary Street-Level Sequences* (MSLS) geo–localisation benchmark, restricted to the 23 cities with the most coverage. Images belonging to the same drive sequence never appear in both train and validation sets; this results in 107 188 training images, 6 116 validation images, and 6 116 holdout (test) images.

Backbone and head. We perform feature extraction with a frozen `ViT-B/16` CLIP encoder, pre-trained on LAION-2B. A 2-layer MLP (input (768) -¿ 256 -¿ 3) projects the 768-d `[CLS]` token into a 3-vector, which is then ℓ_2-normalised to $\mathbb{S}^2$. We also use regression on a scalar concentration $\kappa > 0$ and train with the von Mises–Fisher (vMF) negative log–likelihood (**??**).

Optimiser and schedule. We ran the training in two phases: *(i)* a **linear probe** in which the ViT is kept frozen for 40 epochs; *(ii)* a **light fine-tune** lasting 2 further epochs in which we unfreeze the top 6 transformer blocks. In both phases we use AdamW with learning rate $\eta_0 = 3 \times 10^{-4}$, weight decay 10^{-2}, and batch size 64. A cosine decay schedule with 5-epoch linear warm-up gave the most stable convergence in a small grid search.

Data augmentation. To allow greater generalization, we apply some standard data augmentation: `RandAugment(N=2,M=9)`, $^{+}$`ColorJitter` and a 20% chance of Gaussian blur (kernel size 23 px). As mentioned earlier, horizontal flips are *not* used because they would invert the geo–directional cues.

Hyper-parameter selection. We ran a coarse log-grid over $\eta_0 \in \{1, 3, 5\} \times 10^{-4}$ and weight decay $\in \{10^{-4}, 10^{-3}, 10^{-2}\}$ on a 5-fold random subset (20% of the training set each). The chosen values ($\eta_0 = 3 \times 10^{-4}$, $w_d = 10^{-2}$) minimized the validation error by 4/5 times and are reported above.

5.2. Evaluation Metrics

Given a predicted unit vector $\hat{\mathbf{v}}$ and a ground-truth city centre $\mathbf{v}$, geodesic error is computed as the great-circle distance on the Earth's radius $R = 6\,371$ km:

$$\mathcal{E}(\hat{\mathbf{v}}, \mathbf{v}) = R \cdot \arccos(\langle \hat{\mathbf{v}}, \mathbf{v} \rangle). \qquad (3)$$

We report (a) mean error (**ME**), (b) median error (**MedE**), (c) hit-rates HR_d — the fraction of images localised within d km, with $d \in \{25, 100\}$.

Model	**ME**↓	**MedE**↓	HR_{25} ↑	HR_{100} ↑
Linear probe (epoch 40)	672 km	175 km	23.1%	42.0%
+ fine-tune (ours)	**489** km	**103** km	**33.2**%	**49.6**%

Table 3: Validation performance on the MSLS 23-city split. Arrows indicate whether lower (↓) or higher (↑) is better.

5.3. Quantitative Results

Table 3 summarizes the best single-model validation performance.

Training dynamics are illustrated in Figure 3, which shows the negative log-likelihood (NLL), mean and median error, and concentration κ over 40 epochs. Note that all curves flatten after approx. 30 epochs with no subsequent divergence between training and validation, suggesting little overfitting under the current capacity.

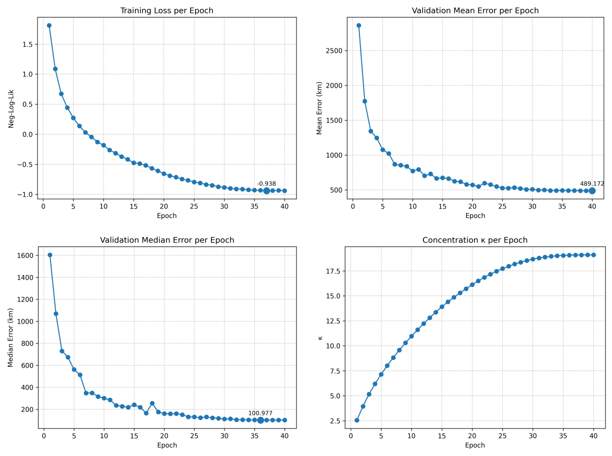

Figure 3: Training curves for NLL, mean/median geodesic error, and estimated concentration κ. Bold markers denote the checkpoint with the lowest validation ME.

Ablation: probe vs. fine-tune. Fine-tuning just 30% of the ViT parameters yields a 27% relative improvement in ME (Table 3). We experimented with unfreezing deeper layers but observed diminishing returns coupled with overfitting, hence we retain the light fine-tune as a sweet-spot.

5.4. Qualitative Analysis

Figure 4 visualises Grad-CAM activations for three random frames per city. We visualize the *norm1* layer for the final block of the model backbone. The aim here is to try and determine what the model is really looking at to identify different cities, given one of our goals was to get an insight into strategies used by top GeoGuessr players.

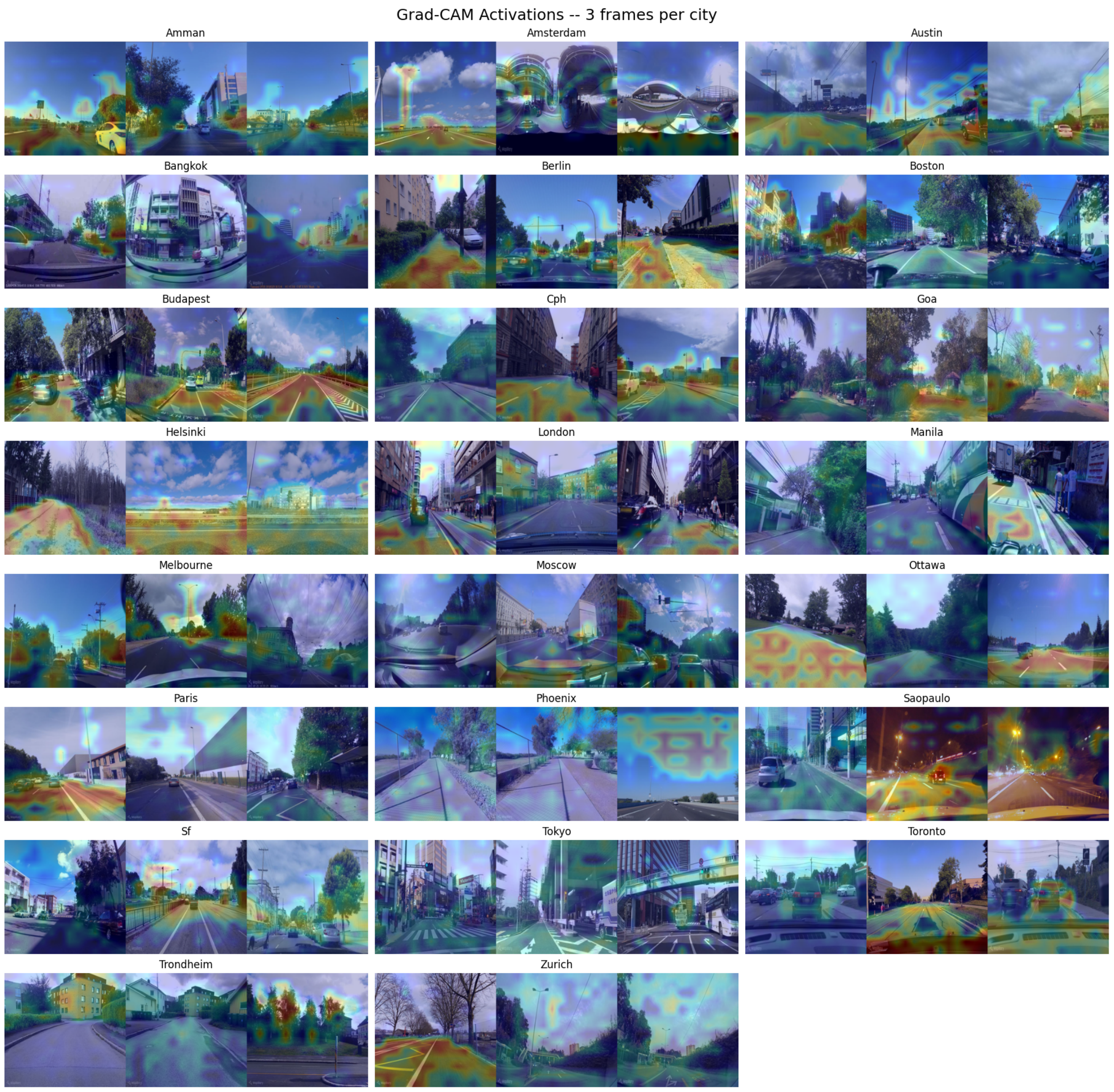

Figure 4: Grad-CAM heat-maps illustrating the image regions that most influence the predicted geolocation. Best viewed digitally.

The network consistently attends to horizon lines, lane markings, license plates (when cars are close enough), and distinctive skyline silhouettes (e.g. the Boston financial district) across cities.

Interestingly, the model seems to pay attention to vegetation in tropical cities including Bangkok in Manila. Electric pylons and streetlamps are also common focuses for the model, as seen in Amsterdam and Berlin. In London and Tokyo, we see the classic buses associated with these cities are highlighted, as they provide a clear hint to the model, while other cars and bikers are not, as these are common features in other cities as well.

These are intuitive strategies for GeoGuessr, offering insight as well as reassurance that our model is indeed learning the right thing.

On the other hand, we also noticed some areas for improvement: in several images, in Moscow and Toronto, the model seems to be paying a lot of attention to the visible portion of the car's interior at the edges of the photos. In retrospect, this is likely because, even though we had split our dataset by sequence IDs, it is likely that the same car's dashcam could have contributed different sequences for a city, contributing to the model overfitting on that property. Is it also possible that the model is picking up on the varying popularity of different makes and models of cars across these cities, with some cars being more popular in specific cities.

Overall, qualitatively, we are able to find sufficient evidence that our model is converging on intuitive features for the task at hand. For the future, we can learn that overfitting can be further avoided by cropping the bottom 20% of images, so that car interiors are not visible in any images.

Failure cases. Despite the above, the model struggles with *night-time imagery*, especially in generic highways lacking city-specific cues. These images frequently had the highest rates of error. We can intuitively understand this as these images give the model little to work off of: texture information is not easily visible in compressed nighttime imagery, and when the scene is of a random, generic highway, the model does not have much to go off of in terms of architectural cues. The model should still be able to leverage other information such as road signage though, so increasing the size of the dataset along this specific category may be helpful.

5.5. Overfitting Discussion

The smooth validation curves and monotonic increase of κ (Figure 3d) indicate that the model learns to assign higher confidence as the geodesic error falls, without memorizing its training data. Dropout was used in the MLP layers, along with image augmentation for the input images, to help with regularisation.

5.6. Summary

The proposed CLIP + vMF head achieves a **median error of 103km** on the MSLS 23-city split. Qualitative inspection corroborates our hypothesis that the network learns semantically meaningful cues (road markings, skylines), while failure cases highlight avenues for improving texture-less scenes, specifically in the nighttime.

6. Conclusion & Future Work

We tackled the single–image geolocation problem by casting it as metric regression on the unit sphere and training a lightweight von Mises–Fisher (vMF) head on top of a CLIP ViT–B/16 backbone. A two–stage schedule *linear probe* followed by a short, partial fine–tune of the last six transformer blocks proved to be the sweet spot between bias and variance. This configuration achieved our best numbers: **489 km mean error**, **101 km median error**, and a **49.6% hit-rate within 100 km**. We found that increasing the backbone's trainable depth beyond six blocks or unfreezing it from the very first epoch led to rapid overfitting without appreciable gains on the validation set.

Qualitatively, Grad-CAM visualisations revealed that the model attends to stable geo–informative cues, such as road layout, vegetation type, sky colour gradients, and region-specific features such as regional buses and the shapes of electric pylons, as opposed to brittle artefacts like JPEG blocks. Error analysis suggests that night-time scenes frequently have the greatest amount of error, underscoring the model's reliance on high–frequency texture cues that are absent under poor illumination.

Future Work. Given more time and compute we would try extending the training set beyond 23 cities to a truly global coverage, possibly via labelled Flickr images. We would also like to try adding a retrieval flow following the neural network pass, similar to PlaNet's approach, to help further cover the gap. The neural network component could be especially helpful for regions with sparser coverage, where reasoning will be important, while retrieval could help improve accuracy in areas with high coverage, such as large cities. It would also be valuable to try different model backbones, and consider experimenting with RL approaches, as well as newer techniques like *Mixture-of-Experts* routing to capture region-specific features.

We feel that this area is rife for future exploration, and the problem continues to be interesting with scalable solutions waiting to be found.

References

[1] G. Astruc, N. Dufour, I. Siglidis, C. Aronssohn, N. Bouia, S. Fu, R. Loiseau, V. N. Nguyen, C. Raude, E. Vincent, L. Xu, H. Zhou, and L. Landrieu. OpenStreetView-5M: The Many Roads to Global Visual Geolocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21967–21977, June 2024.

[2] A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

[3] M. L.-A. P. G. Y. K. J. C. Frederik Warburg, Soren Hauberg. Mapillary street-level sequences: A dataset for lifelong place recognition, 2020.

[4] L. Haas, M. Skreta, S. Alberti, and C. Finn. Pigeon: Predicting image geolocations, 2024.

[5] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[6] K. Mardia and P. Jupp. *Directional Statistics*. Wiley, 2000.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[8] N. N. Vo, N. Jacobs, and J. Hays. Revisiting IM2GPS in the Deep Learning Era. *CoRR*, abs/1705.04838, 2017. arXiv:1705.04838.

[9] T. Weyand, I. Kostrikov, and J. Philbin. *PlaNet - Photo Geolocation with Convolutional Neural Networks*, page 37–55. Springer International Publishing, 2016.