1. Supplementary

This article forms the supplementary material for our paper which aims to provide a better insight into our methods, and also provide additional details which we were unable to include. Here, we expand upon the following :

1. Additional Ablation Studies
   (a) Ablation on Number of Scenes
   (b) Ablation on Number of Hierarchies
   (c) Efficiency and Memory Requirements
2. Analysis of the Google World Streets 15k dataset
   (a) Lorenz Curves
   (b) Gini Coefficient
   (c) Examples of Image Localizations
3. Review of Baselines
   (a) Encoder Baselines
   (b) Pre-existing Methods
4. Implementation Details
   (a) Hyperparameters
   (b) Augmentations
5. Qualitative Analysis

1.1. Additional Ablation Studies

1.1.1 Ablation on Number of Scenes

Previous works [6] [7] have already emphasized the importance of having scene-type information in training labels for geo-localization, so this was not the focus of our work. However, we provide results to both show the necessity of these labels, as well as the optimal number to use. On both IM2GPS3k [10] as well as YFCC26k [9], we found diminishing returns when using 365, with a drop of 0.2% 1KM accuracy, as seen in Table 1.

Table 1. **Ablation Study on Number of Scenes.** We show the affect that the number of scenes has on accuracy. Using 16 scenes outperforms every other option on nearly all metrics.

Dataset	# of Scenes	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS3k [10]	0	11.8	30.4	46.2	58.3	77.6
	3	12.0	31.7	47.0	59.8	78.4
	16	**12.2**	**32.0**	**47.9**	**60.5**	**79.8**
	365	11.9	31.8	47.2	58.5	78.6
YFCC26k [9]	0	8.0	19.8	30.1	44.6	62.2
	3	8.4	20.5	31.0	46.0	64.8
	16	**8.7**	21.4	**31.6**	**47.8**	**66.2**
	365	8.5	**21.6**	30.2	46.4	64.9

Table 2. **Ablation Study on Number of Hierarchies.** We show our results when varying the number of hierarchies used during training.

Dataset	# of hierarchies	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS3k [10]	1	9.8	29.6	41.1	56.4	73.5
	3	12.8	34.5	**46.1**	**61.5**	**76.7**
	5	13.4	34.4	45.4	61.1	76.1
	7	**14.3**	**34.8**	45.7	61.3	76.0
YFCC26k [9]	1	6.7	18.2	29.0	45.2	64.0
	3	10.1	**24.3**	34.7	**50.1**	**67.8**
	5	10.2	24.1	**34.8**	50.0	67.7
	7	**10.8**	23.5	34.0	49.3	67.4
GWS15k	1	0.0	0.9	5.7	21.8	44.0
	3	0.2	1.3	7.9	**25.4**	**49.4**
	5	**0.6**	**1.7**	**8.1**	24.3	48.0
	7	0.2	1.0	6.9	22.7	46.2

1.1.2 Ablation on Number of Hierarchies

While previous works utilized geographic hierarchies, they were either used separately [10] or limited to only using 3 levels of specificity [6, 7]. We are the first to use more than 3 hierarchies in a combined manner. As mentioned in the main paper, our hierarchies are defined by limiting the number of training images in an S2 cell. All hierarchies have a minimum threshold of 50 images, while the maximum number of images is anywhere from 25000 to 500 depending how geographically fine-grained each class must be, specific values can be found in Table 4. We use at most 7 hierarchies but experiment with 1, 3, 5, and 7 on the testing datasets Im2GPS3k [10] and YFCC25600 [9]. Each model is trained only with the number of classifiers specified by the number of hierarchies. The ultimate goal of geo-localization is to predict the location of an image as accurately as possible. With this in mind Table 2 shows that adding more hierarchies improves geo-localization accuracy at the 1KM scale by as much as 1.5% over the established 3 hierarchies. However, if one were not wanting to find the exact location, but instead the country or continent, then it seems that 3 hierarchies shows the best result. It appears that introducing extra fine-grained geographic classes causes our model to focus on extracting features to predict an image's location as precisely as it can at the cost of not finding features that correspond to coarser geographic hierarchies.

1.1.3 Efficiency and Memory Requirements

We provide a comparison of parameters and GMACs with the previous state-of-the-art Translocator [7]. We can see that our method with a Vision Transformer [1], which is the encoder used in Translocator, has significantly fewer parameters and GMACs than Translocator. Our best model which uses the SWIN [4] has a comparable number of parameters but fewer GMACs than SOTA.

Table 3. **Efficiency and Memory Requirements.** We show the Parameters and GFLOPs of our method compared to previous SOTA. Results denoted with * are using our recreation of the given model.

Model	# of Parameters	GMACs
Translocator* [7]	340M	78.67
Ours (ViT)	128M	19.73
Ours (Swin)	344M	53.1

1.2. Analysis of the Google World Streets 15k dataset

In the main paper, we discussed how previous world-wide geo-localization datasets focused heavily on tourist heavy *landmarks*, ignoring systems' ability to localize more common, everyday scenes. Additionally, we showed our Google World Streets 15k is designed to capture more of these scenes, by taking random Google Street View snapshots based on a landmass-based weighting metric, rather than pulling from photography databases.

1.2.1 Lorenz Curves

A Lorenz curve is a technique used to measure the distribution of some resource. [5] While typically used for income inequality, we can use it to demonstrate the distribution of images globally within each dataset. In our case(Fig. 1), the x axis represents the cities with the bottom x% of total images, with the y axis representing the cumulative number of images. A perfectly equal distribution would be represented by the line $y = x$. While both both datasets contain some level of inequality (as larger metropolitan cities are both larger and have more images available), the top 1% of cities within IM2GPS3k have significantly more images than even the top 5%.

1.2.2 Gini Coefficient

The Gini coefficient G is a measure of inequality within some frequency distribution [2]. It is an estimation of the area under a distribution's Lorenz curve vs the ideal case ($y = x$), thus providing a rigorous calculation of equality. Again, in our case, this distribution will be the number of images per city. Let i and j represent two cities. x_i represents the number images in city i. $\overline{x}$ is the average number of images per city, across all cities, serving as a normalization term. Then, we calculate G as follows.

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \overline{x}} \tag{1}$$

In the ideal case, every city has the same number of images, and therefore the difference between all cities is zero. Therefore, the resulting G would also be 0. Performing this calculation on each of our validation sets, we find IM2GPS3K's and YFCC26k's coefficients to be 0.60 and 0.73, respectively. GWS15k's coefficient is 0.51, an improvement of nearly 0.10

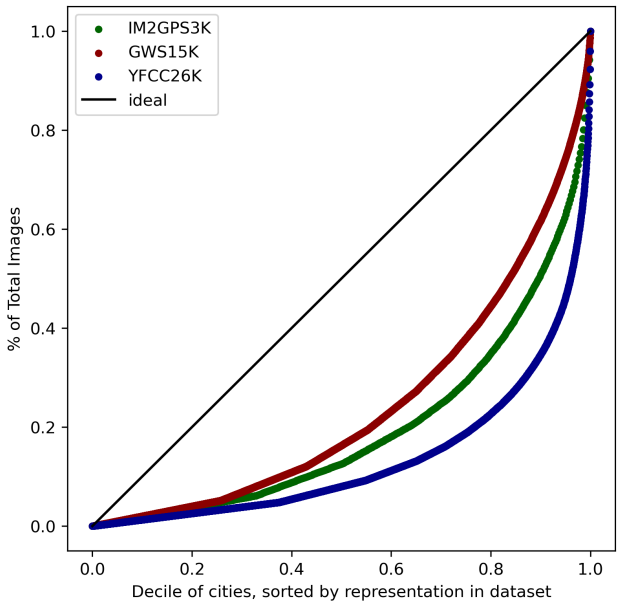

Figure 1. A comparison of the Lorenz curves of each validation dataset. The x-axis represents *the botton x decile of cities, ordered by number of images* while the y-axis represents the number of images that decile contains. The black line represents perfect equality, and therefore the closer to this line, the better. While all three datasets contain some level of inequality, we see the curves of IM2GPS and YFCC rise sharply near the end, implying the most represented cities makeup a very large percentage of the dataset.

1.2.3 Examples of Image Localizations

In the following subsection, we detail a number of example localizations from our GWS15k dataset, as well as an example of failure cases. These are shown in Figures 2, 3, 4, 5, 6 and 7. As expected, strong geo-localization is commonly realized in images that contain noticeable landmarks, architecture, and geography. However, GWS15k also challenges geo-localization systems with common, everyday locations. These types of locations are also represented in locations correctly geo-localized by our model. Examining failure cases, we can see that we still struggle with locations that contain almost no man-made structures or structures not specific to a location–such as a playground.

1.3. Review of Baselines

- **Vision Transformer (ViT)** [1] This architecture, inspired from Natural-Language Processing, breaks an

Table 4. **Training parameters for our model**

Hyperparameter	Value
Batch-size	512
Epochs	40
Optimizer	SGD
Learning Rate	0.01
Momentum	0.9
Weight Decay	0.0001
Hierarchy Losses	Cross-Entropy
Scene Loss	Cross-Entropy
Scheduler	MultiStepLR
Milestones	[4, 8, 12, 13, 14, 15]
Gamma	0.5
Maximum Images per Class	[25000, 10000, 5000, 2000, 1000, 750, 500]
Minimum Images per Class	50
Classes per Hierarchy	[684, 1744, 3298, 7202, 12893, 16150, 21673]

image into non-overlapping squares, called patches, and feeds them through multiple layers of self-attention. For our baselines with this encoder, we use the ViT-B architecture pre-trained on ImageNet 21k [8] and train it on MP-16 [3]. For classification we use the cls token output and feed it into 3 classifiers (one for each geographic hierarchy) and a classifier to predict the scene label as an ancillary task. The results of this are show in Table 5 of the main paper.

- **Shifted Window Transformer (Swin)** [4] Building off of ViT, Swin Tranformers re-think the self-attention step and instead perform attention only within specific windows that shift at different layers. Swin also performs a patch merging operation which lets the model learn multi-scale features. We use this architecture pretrained on ImageNet 21k [8] and train it on MP-16 [3] as we did with ViT. Swin, however, does not use a cls token but instead outputs a feature map of size 7×7. Therefore, we average pool the feature map to get one set of features, which is passed to the geographic and scene classifiers. These results are also in Table 5 of the main paper.

1.4. Implementation Details

1.4.1 Hyperparameters

Our model is trained for 40 epochs with a batch-size of 512. We utilize Stochastic Gradient Descent with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 0.0001. Our encoder is pretrained on ImageNet [8] and we use Cross-Entropy for all of our losses. We outline the training parameters for our model in Table 4.

1.4.2 Augmentations

We utilize the following augmentations during training:

- Random Affine (1-15 degrees)
- Color Jitter (brightness=0.4, contrast=0.4, saturation=0.4, hue=0.1)
- Random Horizontal Flip (probability=0.5)
- Resize (256×256)
- RandomCrop (224×224)
- Normalization

During Evaluation we use:

- Resize (256×256)
- TenCrop (224×224)
- Normalization

TenCrop is an augmentation technique that, given an image, returns the center and corner crops as well as the horizontally flipped version of each of those crops.

1.5. Qualitative Analysis

In figure 8 we show the locations that our model predicts within each distance threshold for Im2GPS3k, YFCC26k, and GWS15k. The overall dataset distributions are shown for reference. We observe that our method can accurately predict images in locations that other datasets leave out. This shows our model's capabilities better by ensuring we test on images all around the Earth and aren't biased towards North America and Europe.

In figures 9, 10, 11, 12, 13, and 14 we provide attention maps from Im2GPS3k [10], YFCC26k [9], and GWS15k. We provide one success and one failure case for each of these datasets. The attention maps are created by looking at the first attention head in the final layer of cross-attention in our Hierarchy Dependent Decoder. This shows the relation between the decoder queries and the image patches from our encoder. The attention maps for every hierarchy and scene query are shown as well as the original image in the top left for reference.

Figure 2. A random sample of GWS15k images our system correctly geo-localizes within 1KM. While well-known landmarks are greatly represented in this sample, we can also see more common locations, such as parks and a city market.

Figure 3. A random sample of GWS15k images our system correctly geo-localizes within 25KM. In this sample, we can begin to see more natural and non-urban locations. Here, our system is able to correctly identify the city of neighborhoods, as well as the rough location of rural highways with interesting structures.

Figure 4. A random sample of GWS15k images our system correctly geo-localizes within 200KM. Here, we begin to see more challenging images. While we have two images from a city street, every other image is on an exurban road or a rural highway. Nevertheless, we can notice geography such as gravel quarries or plateaus that assists our system in identifying these locations.

Figure 5. A random sample of GWS15k images our system correctly geo-localizes within 750KM. As this sample includes images of which the country was correctly determined, we can see examples of architecture specific to nations, but not necessarily regions. These types of structures often include types of landposts, or road signs.

Figure 6. A random sample of GWS15k images our system correctly geo-localizes within 2500KM. As these are images of which only the continent could be determined, nearly all samples of are rural locations. These images show little differentiable geography, but enough fauna or examples of architecture to determine the continent.

Figure 7. A random sample of failure cases, where our system's geo-localization error was over 3000KM.

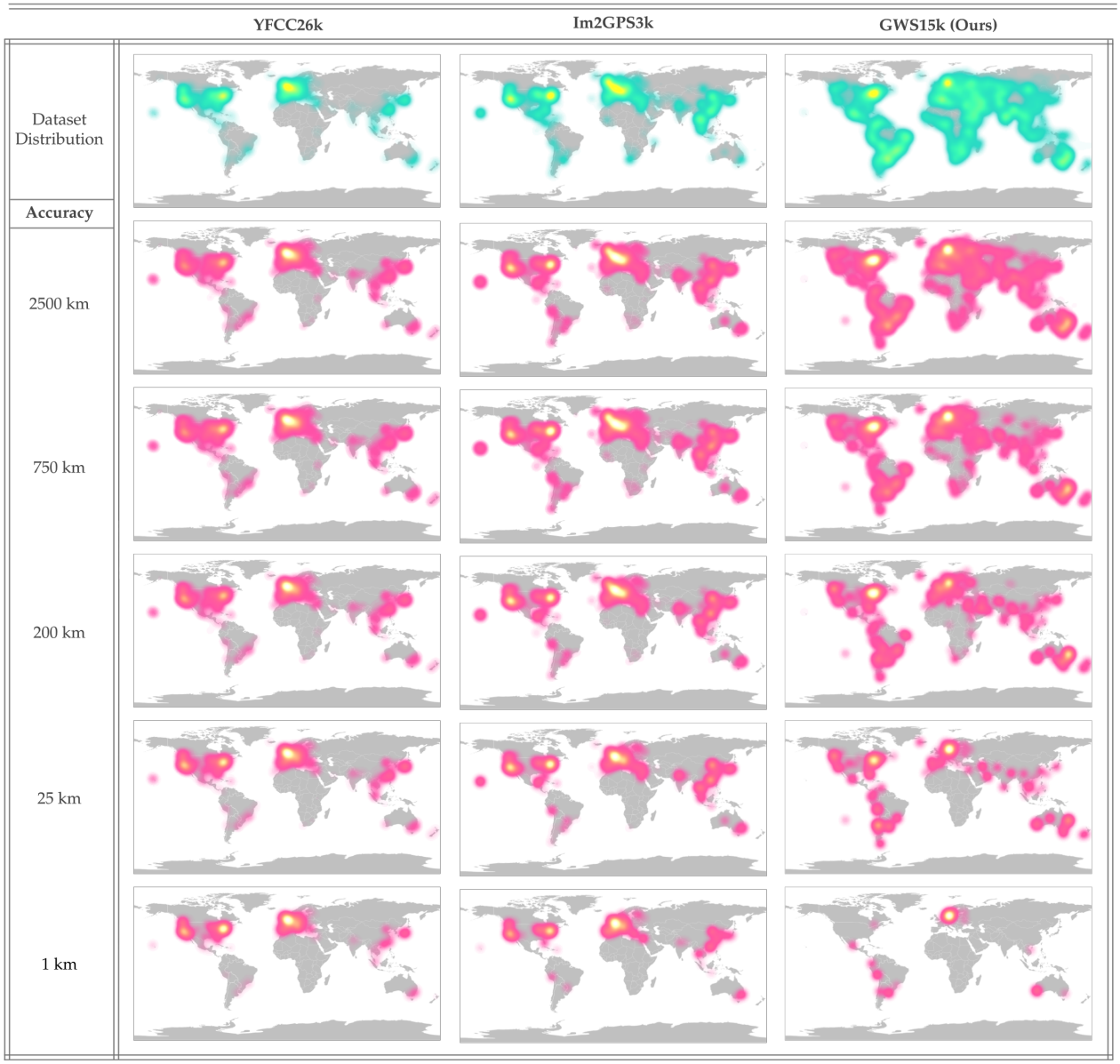

Figure 8. A visualization of the distribution of our model's correct predictions for different accuracy thresholds and testing datasets. By observing the predictions of our model on GWS15k we can note that we are capable of identifying places that are underrepresented in YFCC26k and Im2GPS3k. It is important to note that although our model's performance is being limited by the training set–whose distribution is comparable to that of YFCC26k–we accurately geo-localize images in areas that are not densely covered with high precision (e.g. Western Asia, South America, and Central Australia).

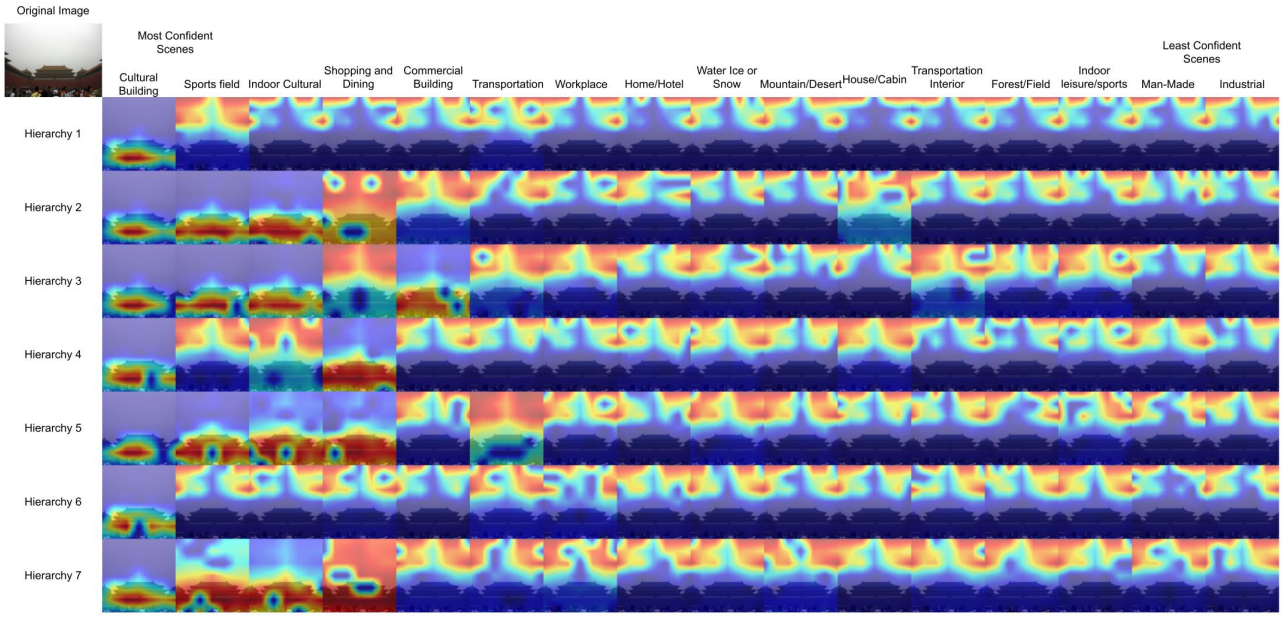

Figure 9. A visualization of all the attention maps for an image of The Palace Museum in Beijing from Im2GPS3k that we predict within 0.32 KM. We see that the left most column, which represents the query we use for classification, has a far more direct attention map then the incorrect scene queries.

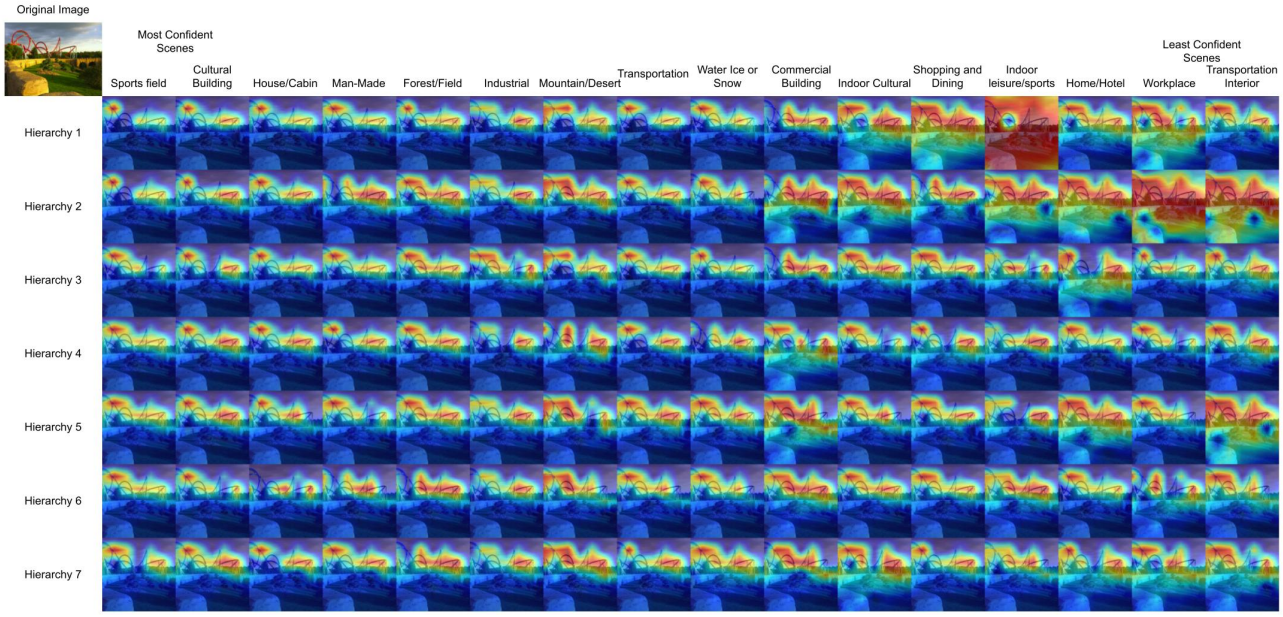

Figure 10. A visualization of all the attention maps for a failure-case image from Barcelona, Spain in Im2GPS3k that we mispredict by 5284 KM. We see that nearly all of the queries are focusing on the roller coasters seen in the background. Our model was not able to find features in the image specific enough to a scene or hierarchy in order to geolocalize it.

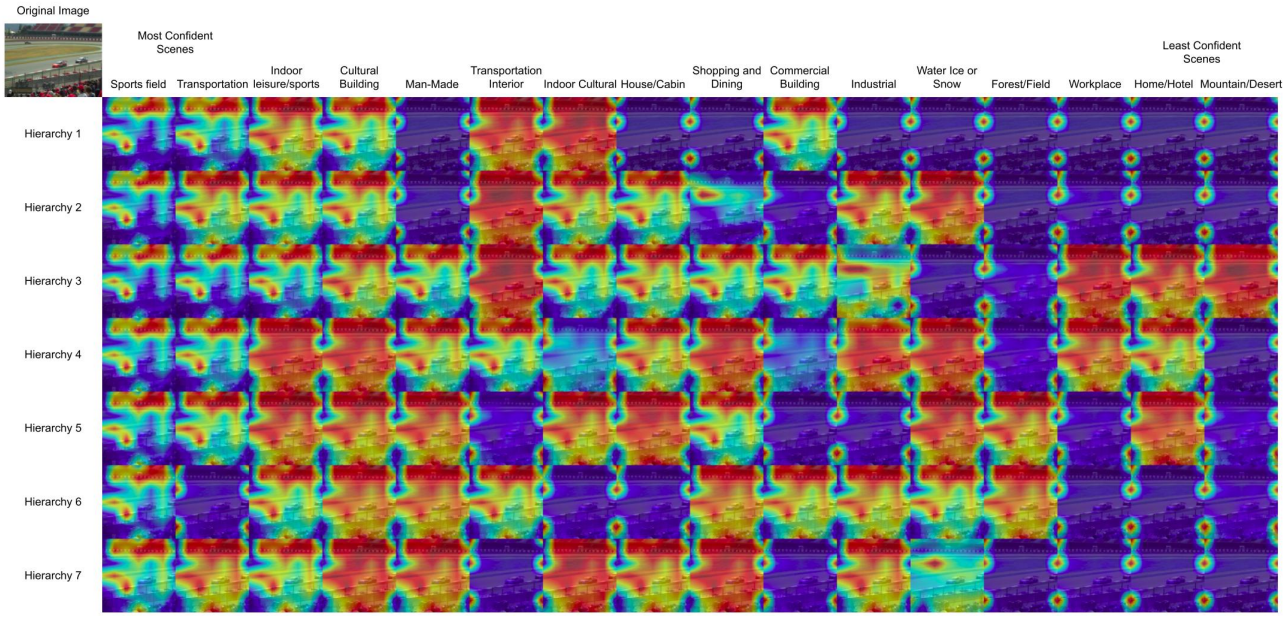

Figure 11. A visualization of all the attention maps for a success-case image at Catalunya Circuit in Barcelona from YFCC26k that we predict within 0.75 KM. Note that the queries used for classification focus on the stands in the background and part of the race track, while the less confident scenes focus either on the corners or generally about the entire image.

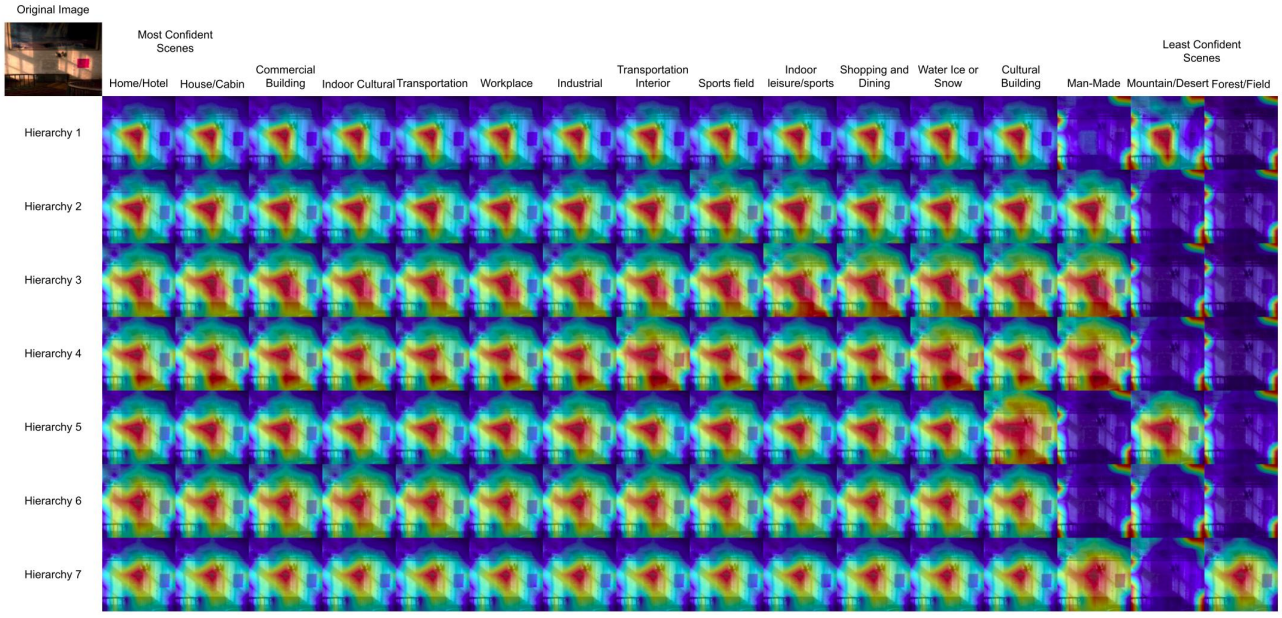

Figure 12. A visualization of all the attention maps for a failure-case image from Anshan, China in YFCC26k that we mispredict by 8442 KM. We see here that almost all scenes show identical attention maps no matter how confident we are in that scene's prediction. Note that this image is also an indoor image of a wall inside this building so we expect this to be an especially difficult image to localize unless images of the same wall exist in the training set.

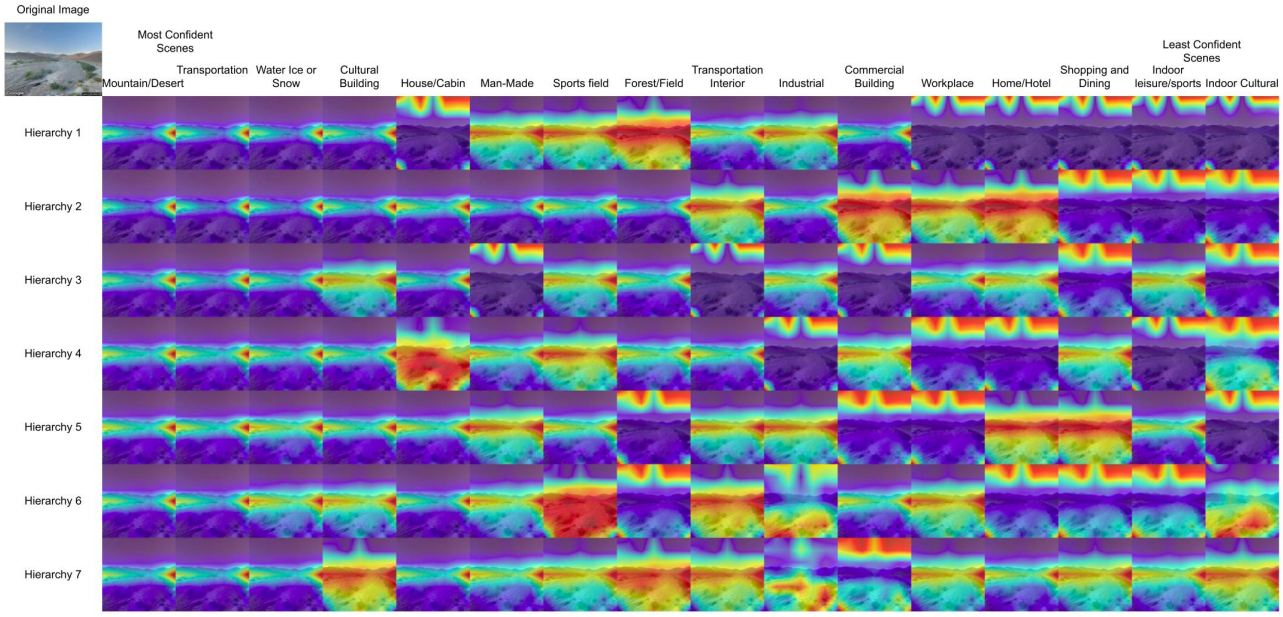

Figure 13. A visualization of all the attention maps for a success-case image from Samail, Oman in GWS15k that we predict within 7.3 KM. We see that the most confident scene queries are consistently focusing on the mountains in the background while the less confident queries do not, showing that our scene selection process helps our model get the best features for geo-localization.

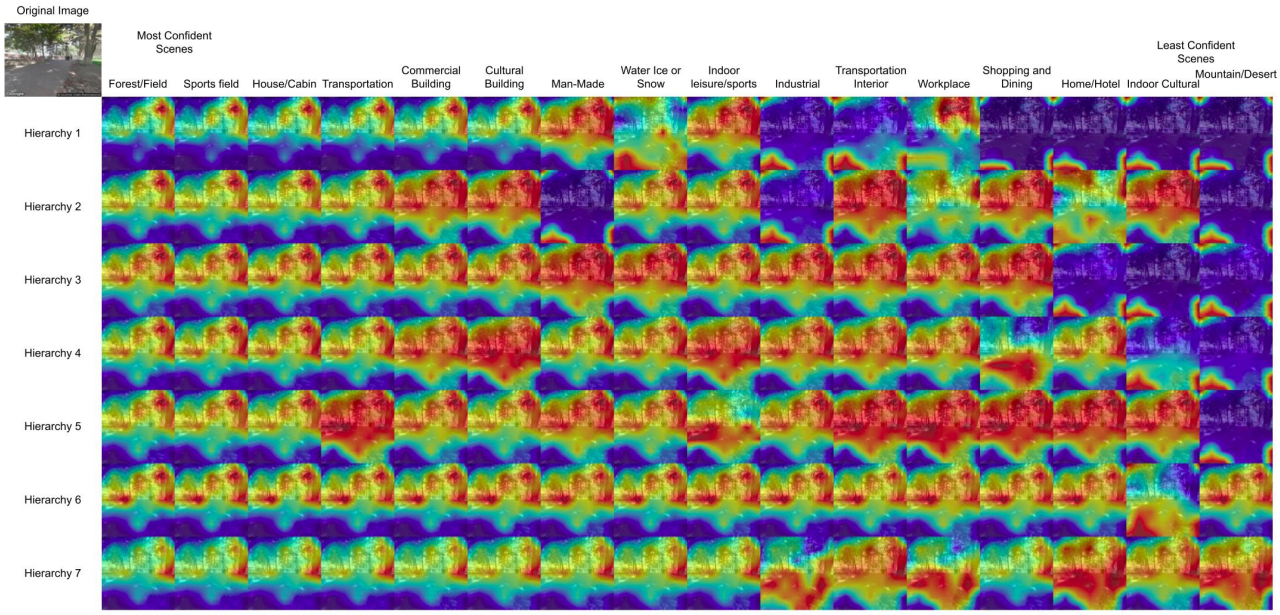

Figure 14. A visualization of all the attention maps for a failure-case image from Mashal University, Afghanistan in GWS 15k that we mispredict by 3490 KM. We see that in this case the queries we would use (the leftmost column) share similar attention maps to the less confident scenes, meaning we could not distinguish this image's features well enough.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[2] RR Gates. Mutability and variability. *Botanical Gazette*, 47(6):476–477, 1909. 2

[3] Martha Larson, Mohammad Soleymani, Guillaume Gravier, Bogdan Ionescu, and Gareth JF Jones. The benchmarking initiative for multimedia evaluation: Mediaeval 2016. *IEEE MultiMedia*, 24(1):93–96, 2017. 3

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 3

[5] Max O Lorenz. Methods of measuring the concentration of wealth. *Publications of the American statistical association*, 9(70):209–219, 1905. 2

[6] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018. 1

[7] Shraman Pramanick, Ewa M Nowara, Joshua Gleason, Carlos D Castillo, and Rama Chellappa. Where in the world is this image? transformer-based geo-localization in the wild. *arXiv preprint arXiv:2204.13861*, 2022. 1, 2

[8] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 3

[9] Jonas Theiner, Eric Müller-Budack, and Ralph Ewerth. Interpretable semantic photo geolocation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 750–760, 2022. 1, 3

[10] Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 2621–2630, 2017. 1, 3