Supplementary Material for LIP-Loc: LiDAR Image Pretraining for Cross-Modal Localization

Sai Shubodh Puligilla [*†] Mohammad Omama[‡] Husain Zaidi[§] Udit Singh Parihar[†]
Madhava Krishna[†]

1. Qualitative Results & Semantic Breakdown of KITTI360

Fig 1 and Fig 2 demonstrate the qualitative results of 2D to 3D localization and 3D to 2D localization respectively on KITTI-360 dataset using the best model LIP-Loc. The 3D scans are shown in top view. As can be seen in Fig 1, the 3D scans that we are able to predict are very close to ground truth scans, which can help in navigation in an environment where 3D information is not available at test time. Similarly, Fig 2 shows that the RGB images retrieved via 3D-2D localization are similar to ground truth, thus this could further result in downstream application like finding a finer pose through perspective-n-point algorithm.

Let us discuss about semantic breakdown of KITTI-360 dataset now. KITTI-360 is a diverse suburban dataset with 37 label classes, including 24 "instance" classes and 13 "stuff" classes. They define a category and within a category come many classes. For example, the category "flat" contains "classes" like road, sidewalk, parking etc; construction contains building, garage, wall, fence etc. They additionally do a statistical analysis over the distribution of the semantic labels, through which they plot 2D semantic labels over frames and 3D semantic labels over points and bounding boxes. When it is done over frames or points, they find that the highest distribution is of classes vegetation, sky, terrain, car and road while when done over bounding boxes reflects that the highest distribution is for classes car, pedestrian, rider, building and bicycle. There are some predominantly "downtown" scenes, i.e. those with buildings/houses, and many objects like trees, bicycles are common, such as sequences 0000, 0002, 0009. There are also some predominantly "highway" scenes, i.e. i.e. those with open areas, continuous vegetation, roads and cars such as 0003, 0004, 0005. Fig-5 of main paper reported recall@1 values on these sequences. Overall, without any training, our "Zero-shot LIP-Loc" performs well in all sets of diverse conditions having a recall of around 0.5 for most sequences and reaching a maximum of 0.658 for 1 sequence. There is no clear correlation between the accuracy on a sequence and its semantic distribution, i.e. whether it is highway or downtown. For example, if we look at highway scenes such as sequences 0003 and 0005 have recall of 0.658 and 0.594, whereas other highway sequences like 0004 have low recall value like 0.470 whereas downtown scene like 0000 has recall of 0.541. This could mean that our "Zero-shot LIP-Loc" model is not learning spurious correlations, in other words, it is not fitting to certain distribution, rather it is learning in a generalized way. As opposed to our baseline AECMLoc which has tested only on 0000 which has one kind of distribution predominantly, we have tested on 6 sequences each of which differs and we get good recall values for each and do not get abnormally poor values anywhere, which suggests that our approach is robust to distribution shift. With that being said, we have to point out that KITTI-360 does not give clear per-sequence breakdown of semantics, and there is a necessity for a benchmark to do thorough analysis and demonstrate the true zero-shot effectiveness of approaches like ours.

2. Architecture: Different Encoders & Bigger Models

Different Encoder	Seq 8	Seq 9
exp_large (resnet)	0.179	0.147
exp_larger (resnet)	0.295	0.309
exp_largest (resnet)	0.484	0.457
trip_larger_vanila (resnet)	0.215	0.232
exp_large (ViT)	0.278	0.260
exp_larger (ViT)	0.547	0.525
exp_largest (ViT)	0.805	0.780
trip_vanila_larger (ViT)	0.279	0.282

Table 1. Recall@1 on Different encoders

In this section, we report additional experiments on experimenting with different encoders and bigger models. We report in the main paper that *vit_small_patch16_224* is the

[*]Corresponding author: p.saishubodh@gmail.com, Project page: https://shubodhs.ai/liploc
[†]Robotics Research Center, KCIS, IIIT Hyderabad
[‡]University of Texas at Austin
[§]Microsoft

Figure 1. Visualization of 2D to 3D localization

Bigger Model	Seq 8	Seq 9
exp_largest_resnet101	0.477	0.462
exp_large_resnet101	0.178	0.152
exp_largest_vit_base_patch16_224	0.777	0.720
exp_large_vit_base_patch16_224	0.238	0.230

Table 2. Recall@1 on Bigger Models

Combined Models	Seq 8	Seq 9
exp_largest (ViT)	0.805	0.780
exp_combined_vit (thresh 50)	0.817	0.741
exp_combined_vit (thresh 100)	0.785	0.758
exp_combined_vit_fewshot	0.811	0.773
exp_combined_vit_base_patch16_224	0.827	0.805

Table 3. Recall@1 for model trained on both KITTI and KITTI-360 and inference on KITTI-360 (no overlap): test on 0000 of KITTI-360, while train on the rest.

final model we have chosen. Here we discuss more on why we have chosen that model and what results we have obtained for other models. Please do note that in this supplementary too, wherever not explicitly mentioned, we are referring to ViT's model of *vit_small_patch16_224* model and ResNet's model of *ResNet50*.

Table 1 reports recall values on different encoders. We observed in our experiments that ViT models have a significant accuracy improvement over ResNet. To ensure the comparison is fair, we pick models which have roughly same number of parameters, i.e. *ResNet50* which has 25M parameters (25,557,032) and *vit_small_patch16_224* which has 22M parameters (22,050,664). Even with 3M less parameters, we notice a rise of over 30% accuracy in `exp_largest` case. This pattern can be observed in training over smaller sequences too, such as `exp_larger` and `exp_large`. Even in the triplet vanilla case, we can see a marginal 5% improvement, clearly demonstrating that the edge of ViT over the standard ResNet models.

In Table 2, we report the recall values of bigger models such as *resnet101* and *vit_base_patch16_224*. Although these models have significantly higher parameters, such as 87M for the latter, we do not observe any much change in accuracy. In fact, it dropped marginally. This could be due to the fact that localization datasets are much smaller com-

Figure 2. Visualization of 3D to 2D localization

Figure 3. Semantic Breakdown of KITTI-360 evaluation sequences: The top row represents downtown with cars, buildings whereas bottom row represents highway with more greenery, wide roads. Our "Zero-shot LIP-Loc" model performs well in these diverse conditions without even being trained on this data.

pared to internet-scale datasets like CLIP and bigger models result in overparametrization, thus dropping accuracy.

Does adding more data improve accuracy for these bigger models? In the main paper, we have discussed the standard train-test setting, where we tested on 0000 sequence of KITTI-360 while its training was on rest of sequences of KITTI-360 (0000) or KITTI (Seq 8 and Seq 9), this was "LIP-Loc". Other setting was when we trained on KITTI data and evaluated on KITTI-360, called as "Zero-shot LIP-Loc". `exp_combined` refers to the a third setting, where we train on all sequences of KITTI and KITTI-360 excluding test sequences of KITTI-360 (i.e. 0000) and KITTI (i.e.

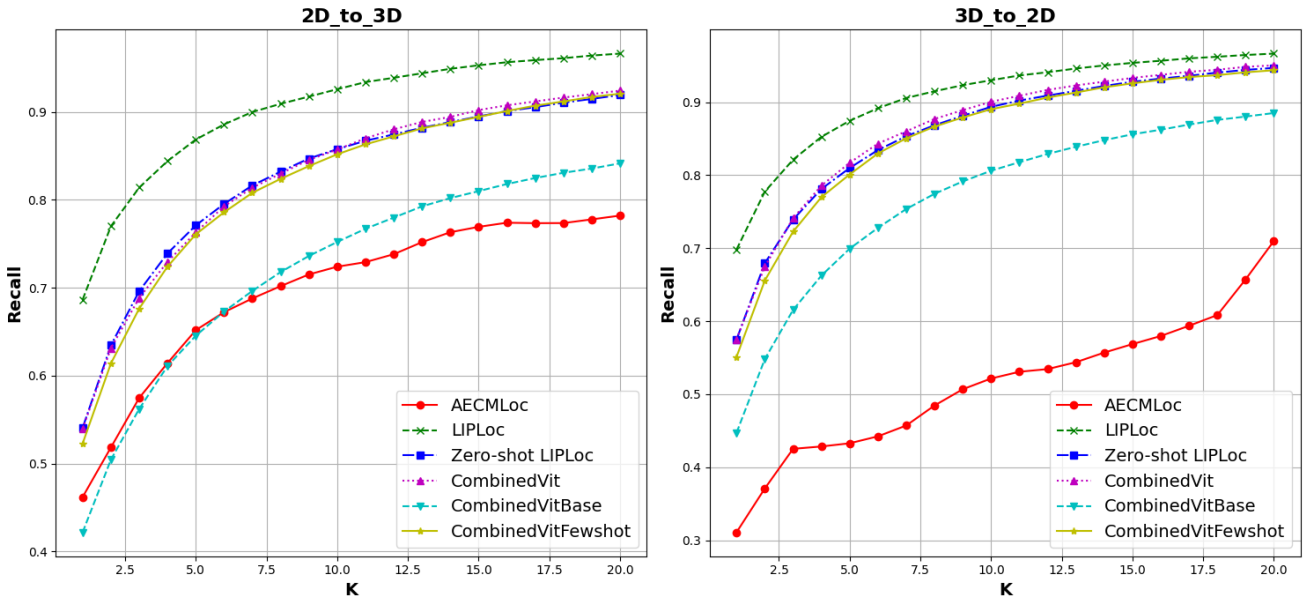

Figure 4. Recall@K curves on KITTI-360 sequence 0000: Combined Models comparison with our best models and baseline AECMLoc

8 and 9) on which we test. To get back to our question of whether adding more data will improve accuracy for bigger models, see last row of Table 3 whose accuracy improved over `exp_largest_vit_base_patch16_224` of 2 by 5%. This further reaffirms that if we scale the model, we need to scale the data in order to improve the accuracy. Do note that this fact is not as established in visual localization as much as in computer vision or language models, it is still an open question as to how much role big data will play for localization, hence these analyses play crucial role.

We also try a few-shot experiment here wherein we give just 1% of data of KITTI-360 when compared to the `exp_combined_vit` experiment. To be clear, `exp_combined_vit` uses all sequences of KITTI and KITTI-360 (excluding test sequences), whereas `exp_combined_vit_fewshot` uses all sequences of KITTI but just 1% of KITTI-360. This is a very captivating result: We receive almost same or marginally improve upon the accuracy as the other experiment despite using significantly very less dataset.

It is also worth noting that the combined experiments don't improve significantly from `exp_largest`, unless we use a bigger model like `exp_largest_vit_base_patch16_224`, which is also 2% improvement. Future experiments have to be done to establish even clearer understanding.

So far, we have discussed evaluation on KITTI dataset. Now let us discuss about evaluation on KITTI-360 dataset by looking at 4. Previously in the main paper, we reported LIPLoc, Zero-shot LIPLoc and AECMLoc (baseline). Here we additionally add the plots of combined experiments, which as described above, merges the training sequences of KITTI and KITTI-360 and trains a single model using full data. Do note that all of our models beat the SOTA AECMLoc. But amongst our models themselves, we rather see ambigious or counterintuitive results.

Firstly to clarify, when we use the term "LIP-Loc", we are referring to standard train/test paradigm, for example when reporting LIP-Loc on KITTI-360, we mean we trained on certain train split of KITTI-360 and evaluating on its test splits; similarly when reporting on KITTI, we mean we trained on train split of KITTI and evaluating on its corresponding test splits. "Zero-shot LIP-Loc" on other hand are trained on full KITTI data but has not seen any KITTI-360 data on which we evaluate. Whereas combined models are trained on train split of KITTI and train split of KITTI-360. Therefore, please keep these nuances in mind when interpreting the result. With that being said, since we are evaluating on test split of KITTI-360, we would expect combined models to significantly outperform Zero-shot LIP-Loc. However, that's not the case here: Amongst the combined models, all the standard ViT model `CombinedVit` and bigger model `CombinedVitBase` and the few shot model `CombinedVitFewshot` give similar recall compared to Zero-shot LIP-Loc and subpar performance compared to LIP-Loc. This further proves that Zero-shot LIP-Loc has generalized very well.

As future work, it will be interesting to see an analysis between zero-shot and few-shot LIP-Loc. This raises many open questions: In computer vision problems which CLIP

deals with, few-shot is clearly defined because it is talking about classification categories. However it is not well defined in visual localization context, which further asserts the necessity of establishment of a well thought benchmark. We encourage the reader to address these open questions and ask the question, "Can big data solve the localization problem?"

CLIP admits that it is not good at task generalization for tasks such as finding close objects in an image or counting the number of objects in an image. Extending our work along the lines of the recent work LiDARCLIP [1] which connects CLIP's embedding space to LiDAR point cloud domain could result in an approach which uses text features to query the right set of points in the LiDAR scan, explicitly identify distance and location of the objects and applying clustering in 3D space to count number of objects (for example) and correlate them with image features to identify the class and appearance of an object. This is especially helpful in extreme low visibility conditions where RGB camera will not work well and LiDAR can help identify objects close to the ego vehicle.

3. Architecture: Hierarchical Design

In models as a follow up to CLIP, many models such as ViCHA [2] propose architectural improvement such as hierarchical alignment. What this essentially proposes is that aligning the two encoders at various levels by adding multiple losses at various layers of text and image encoder. They claim that this helps in convergence faster and results in superior performance. In our experiments, we have hierarchically aligned image and lidar encoders at various layers and report it in first half of the table 4. We have tried two experiments: One that aligns only at final layers, the other that aligns throughout the encoder, as ViCHA argues that aligning at the beginning could result in confusing the model. However, in our experiments we did not observe any noticeable improvement, although ViCHA's observation of alignment at final layers could be verified in the case of visual localization as well.

The second half of the table pertains to the following. In standard CLIP setting, there is no relation between any consecutive images in a batch, as they are just (image, text) pairs. However, in our localization setting, the images are sequential. Therefore, we attempted the question: Can we achieve higher accuracy by grouping together adjacent images and having additional encoder for groups of images which results in secondary loss? The last 3 rows of 4 correspond to these experiments. Do note that these experiments are with *ResNet* architecture. Our results actually deteriorated during our experiments. There is a simple rationale for this: The training of deep models works so well because of randomization of samples in a batch, especially in the case of our batch construction technique. When we contruct groups within the batch and ensure the images within the group are consecutive but groups themselves are random, we are asking for a tradeoff: will the additional hierarchical loss improve accuracy more than reducing randomization will decrease it? We have found in our experiments that the answer is no, even for smalller group sizes such as 4.

This section concludes that sticking to non-complicated architectures works the best since the power of CLIP model subsumes any minor architectural improvement.

Advanced Architectures	Seq 8	Seq 9
exp_large (ViT)	0.278	0.260
hier_align_large_vit (final layers)	0.275	0.258
hier_align_large_vit (all layers)	0.218	0.197
exp_large (resnet)	0.179	0.147
exp_larger (resnet)	0.295	0.309
exp_largest (resnet)	0.484	0.457
hier_group_shuffle_large_resnet	0.170	0.1495
hier_group_shuffle_larger_resnet	0.239	0.212
hier_group_shuffle_largest_resnet	0.378	0.346

Table 4. Architecture: Hierarchical Design

References

[1] Georg Hess, Adam Tonderski, Christoffer Petersson, Kalle Åström, and Lennart Svensson. Lidarclip or: How i learned to talk to point clouds, 2023. 5

[2] Mustafa Shukor, Guillaume Couairon, and Matthieu Cord. Efficient vision-language pretraining with visual concepts and hierarchical alignment, 2022. 5