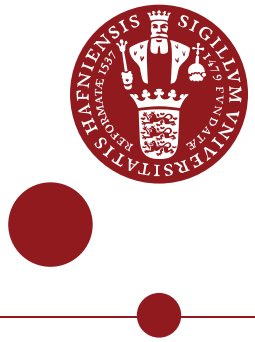
UNIVERSITY OF COPENHAGEN
DEPARTMENT OF COMPUTER SCIENCE



Advanced Topics in Image Analysis

Oscar Koch-Müller

Adapters for Traffic Segmentation

An Empirical Study of Fine-Tuning for Geolocations

Advisors: Abraham George Smith & Jens Petersen

November 2, 2025

Abstract

This project delves into whether using adapters when fine-tuning MobileSAM for specific geolocations within the traffic domain results in segmentations of higher quality, which is done by fine-tuning MobileSAM with three different adapting strategies: LoRA, standard adapter blocks, and no adapter on three different datasets from varying geolocations: the US, Germany and neighboring countries, and India. All experiments were repeated three times for a total of 27 training runs. The results of the experiments show that using no adapter is clearly beneficial for two out of three datasets, while still performing better than the adapter-based approaches on the last dataset on average. The results also demonstrate that applying adapters is not always preferable, despite their popularity, and that results from a vastly different domain do not seamlessly carry over to the traffic domain.

1 Introduction

One major challenge within the field of autonomous driving perception is the context shift across geolocations, which refers to the differing geographic locations of the globe. Previous work, such as [6] by Kalluri et al, has demonstrated these difficulties by evaluating the performance degradation of applying driving perception models to regions other than those for which they were trained on. Unsurprisingly, the results demonstrate a substantial impact on performance due to the context shift of geolocations.
One way to combat such issues would be by fine-tuning a base model for various geolocations, allowing autonomous driving systems to utilize various perception models depending on location. Furthermore, if the base model were to be fine-tuned by a smaller adapter, it would only be necessary to store the base model and one adapter per geolocation fine-tuning, which would reduce the storage space needed, relative to training a new model for each geolocation, assuming many geolocations.

The approach of fine-tuning for specific geolocations is currently being investigated, for example, by Wayve [7], but due to the competitive nature of the autonomous driving systems field, such private actors tend not to disclose the exact architectures and training procedures used, effectively leaving behind a public knowledge gap of how to effectively design autonomous driving perception systems through fine-tuning for specific geolocations. This project aims to address part of the knowledge gap of effectively fine-tuning for geolocations by empirically testing various fine-tuning strategies for various geolocations with the goal of identifying whether using an adapter is beneficial.

More precisely, this project will investigate the following three strategies for fine-tuning the decoder of MobileSAM by Zhang et al [10] in automatic mask generation mode to accurately segment objects common in traffic situations for three different geolocations in terms of dice similarity coefficient (DSC) and thereby address the knowledge gap:

- Low-rank adaptation (LoRA) [5] by Hu et al is an adapter technique that freezes the original weights and introduces small trainable low-rank matrices, which are added to the existing layers. Only these newly added

weights are updated during training. More precisely, the addition of one such low-rank matrix can be formulated as:
$W = W_0 + \Delta W, \quad \Delta W = AB$
Where:
W is the adapted weight matrix.
W_0 is the pre-trained weight matrix.
A and B are small, low-rank matrices that constitute ΔW, which is the update that is learned.

- Standard adapter blocks [4] by Houlsby et al also freeze the existing weights, but unlike LoRA, it applies a learnable down-projection to pre-trained hidden representations followed by a non-linear activation. Then a learnable up-projection is applied, projecting back to the original dimensionality. Finally, a residual connection is included, adding the original hidden representation. More precisely, the addition of one such adapter block can be formulated as:
$h_{out} = h + Uf(Vh)$
Where:
h_{out} is the result of the adaptation.
h is the hidden representation.
V is a down-projection matrix.
U is an up-projection matrix.
$f(\cdot)$ is a non-linear activation.

- No adapter. Without an adapter, we just update the pre-trained model's parameters. Having no adapter is a necessary baseline to assess whether the above adapters are beneficial for the task.

As this project concerns itself with fine-tuning for specific geolocations, three datasets from diverse geolocations were picked: Berkeley DeepDrive [9] by Yu et al, sourced from the US, Cityscapes [1] by Cordts et al, sourced primarily from Germany, but also from neighboring countries, and India driving dataset [8] by Varma et al, sourced from India.

Although applied to an unrelated domain (the medical domain), previous work by Gu et al [3] comparing the same three approaches shows that the standard adapter blocks perform the best when using a similar no-prompt decoder fine-tuning setup for MobileSam for their specified task, although the standard adapter only achieves the better performance by quite a small margin. That work is forming the basis of a working hypothesis for this project, and due to the only marginally better performance of the standard adapter, the working hypothesis is that no adapter strategy will achieve clearly better performance than the others given the setup outlined above.

Code to reproduce results and figures is included in the zip folder. Random seeds were used throughout, ensuring that the results can be reproduced exactly. For details, refer to README.txt in the zip folder.

2 Methods

This section describes what datasets were used, the preprocessing applied, the details of how the experiments were conducted, and finally, how the quality of adapter strategies was evaluated.

2.1 Datasets

The datasets were chosen such that most characteristics are similar except for the geolocation, since the geolocation is what this project varies. Common to all the datasets is the use of a dashcam to capture the images, and the images being sourced in or near cities. Furthermore, all of the datasets are segmentated semantically at the pixel level and have many classes in common.
One example image and annotation for each dataset follows in Figure 1.

2.1.1 Cityscapes

For Cityscapes [1] by Cordts et al, the 3475 finely annotated images were used, which come with 30 annotated classes. Most of the images are sourced from Germany, but a smaller portion is from neighboring countries. The Cityscapes dataset comes with a predefined datasplit designed such that every city is entirely within the training, validation, or test set. The split also ensures that locations from the geographical east, west, north, south, and center, as well as large, medium, and small cities, are evenly split. For the test set, there were no semantic segmentations included.

2.1.2 India Driving Dataset

For India Driving Dataset (IDD) [8] by Varma et al, the IDD Segmentation (IDD 20k Part I) dataset was used, which contains 10,004 images from 182 driving sequences sourced from Bangalore and Hyderabad, as well as their outskirts, annotated with 34 classes. The IDD dataset has a predefined training, validation, and test set that are designed such that every driving sequence is located entirely within one set. For the test set, there were no semantic segmentations included.

2.1.3 Berkeley DeepDrive

For Berkeley DeepDrive (BDD) [9] by Yu et al, only the 10,000 images that are semantically segmented were used. The images are sourced from New York, Berkeley, San Francisco, and the Bay Area and are annotated with 40 classes. The dataset comes with a predefined training and validation set, but the test set is not public. There is no separation of driving sequences or cities among the provided splits.

2.2 Data Preprocessing

Some preprocessing was applied to the data, which is described in the following.

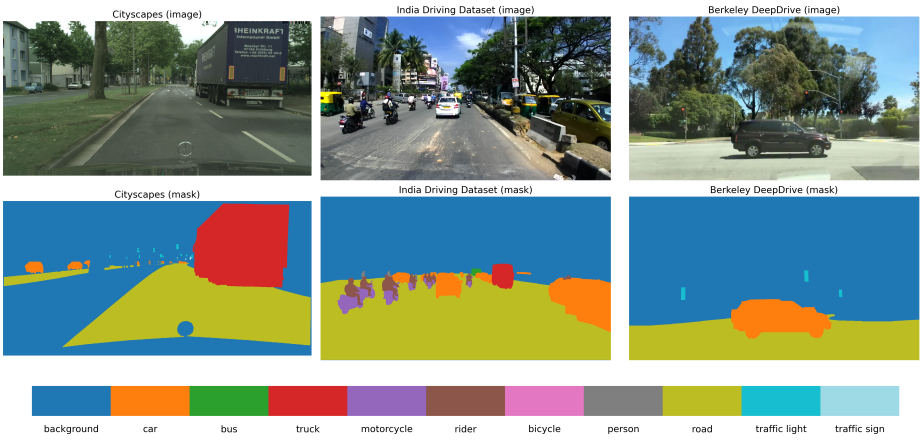


Figure 1: One arbitrarily chosen image and corresponding segmentation for each dataset. The labels are shown after semantic subset selection, as per Section 2.2.2, but before normalisation and resizing as per Section 2.2.3.

2.2.1 Dataset Partitioning

Due to computational limitations, it was decided to only use 750 images for training, 250 for validation, and 1000 for testing per fine-tuning strategy per training run. Since the images in the test sets aren't used during training, it was decided to make the test set relatively large, as a large test set can increase the certainty of the results while not increasing the required compute substantially.

Due to the lack of semantic segmentations for the test sets for all of the datasets, it was decided to discard all data predefined as test data and redo the data splits for this project. For cityscapes, information about which city each image was sourced from is available, and therefore, the separation of cities across the splits was kept intact. Though redoing the splits means the promise of having an even distribution of large, medium, and small cities, as well as geographical east, west, north, south, and center, can no longer be guaranteed.
For India Driving Dataset, the new splits still ensure that all images from any driving sequence are entirely within one of the splits, which is possible as the data is grouped by driving sequences.

2.2.2 Semantic Subset Selection

To ease the task and to align the datasets more, 11 classes were chosen to segment. 10 of these classes were chosen as they're usually of high importance in traffic situations and are explicitly labelled in all three datasets. They are: car, truck, bus, person, bicycle, motorcycle, rider, road, traffic sign, and traffic light. The last class is background, which is defined as everything that isn't these classes. It was decided to treat the background class as any other class, as ensuring that something isn't of importance is also necessary for autonomous driving.

2.2.3 Further Preprocessing

It was decided to use the same normalisation scheme as used in MobileSAM, i.e., normalising each channel independently by subtracting the mean and dividing by the standard deviation. Since the images used in this project are similar to the ones used for training MobileSAM [10] by Zhang et al, it was also decided not to compute means and standard deviations of the data used for this project. Instead, the values computed during training of MobileSAM were reused.
Additionally, all images were resized to a size of 1024x1024 to keep the size constant across datasets, and as previous work has had success with resizing to 1024x1024 when using MobilesSAM, such as [3] by Gu et al. A manual inspection of all three datasets was conducted, and it was found that the resizing does not appear to destroy small objects in the images.
For IDD, the annotations were provided as lists of polygons defining the semantic classes. The polygons were converted to masks to be consistent with the other datasets.

2.3 Metrics

To evaluate the quality of segmentations, the dice similarity coefficient (DSC) is used due to its effectiveness in measuring the overlap of segmentations and annotations. Since there are imbalances in the frequency of classes, for example, background has a pixel share of more than 50% on all training sets, while traffic sign has a pixel share of 0.006% to 0.184% depending on the dataset, and since most of these classes are of similar importance to identify regardless of frequency or size, it was decided to use a macro-averaged DSC (i.e. calculating the DSC scores for each class and then averaging across classes). The choice of macro-averaged DSC ensures that less frequent classes also need to be segmented well to get a high score.
Additionally, the training time for every model was logged as an estimate of the computational resources used, which also should be taken into account when comparing the models.
Due to the small number of training runs (see Section 2.4), it was decided not to compute any measures of statistical significance. Instead, a more heuristic analysis was conducted by analysing the individual data points to assess and interpret the results. More precisely, if one fine-tuning strategy consistently outperforms the others in terms of macro-averaged DSC by a reasonable margin, given approximately the same amount of training time, it will be considered better.

2.4 Experimental setup

3 independent training runs were trained for each adapting strategy for each dataset, totalling 27 training runs. The experimental setup is inspired by Gu et al [3], but modified as described in the following. Their codebase [2] served as a starting point for training models for this project. Gu et al use a loss function that consists of two parts, one being a DSC loss and the other being a cross-entropy loss. This project modified the function to only use the DSC loss, as that will align the loss function more with the metrics described in Section 2.3.

The DSC loss used in this project, inspired by Gu et al, is more precisely passing the model's logits through a sigmoid, after which the predictions are squared before being compared to the one-hot encoded ground truth for the DSC computation. Finally, all DSC values are macro-averaged.
Just as in Gu et al, the training was stopped after at most 200 epochs, an early stopping criterion was used, which stopped the training after 20 consecutive epochs with no increase in performance on the validation set (measured as macro-averaged DSC), the AdamW optimizer was used with a weight decay of 0.1, and a batch size of 4 was used. Unlike in Gu et al, this project used a warmup of 10 epochs (it is 200 in Gu et al) to ensure the learning rate can reach its predefined value before the training terminates. Mobile-SAM was set to automatic mask generation mode for all experiments.

It was decided to only fine-tune the mask decoder of MobileSAM, leaving the encoder as is. The reasoning for not fine-tuning the encoder is that since MobileSAM was trained on natural images, including traffic scene images, the encoder should already be able to extract useful feature representations of the data used in this project. The encoder is naturally not optimised for these traffic scenes, but it is assumed that it is sufficient as is.
Adapter placement and hyperparameter choices are also based on Gu et al and are as follows: For standard adapter blocks, the adapters were added after the multi-head attention in the decoder. For LoRA, the adapters were added to the query and value projection layers within every transformer block of the decoder, and a rank of 4 was used.
All experiments were conducted using an NVIDIA GeForce RTX 5070 TI GPU, an Intel(R) Core(TM) Ultra 7 265KF CPU, and 32 GB of RAM.

3 Results

Figure 2 shows the macro-averaged DSC of every model on the test set. It is noticeable that for Cityscapes and IDD, the three runs with LoRA all achieve a higher macro-averaged DSC than the three runs with standard adapter blocks, but a lower macro-averaged DSC than the three runs with no adapter.
On BDD, LoRA still achieves a higher macro-averaged DSC than standard adapter blocks on average and a lower macro-averaged DSC than no adapter on average, although for BDD, there is no adapter strategy that has all of its runs performing better or worse than all other runs in terms of macro-averaged DSC. That said, the no-adapter strategy achieves a higher macro-averaged DSC than standard adapter blocks for every single run, when not comparing across datasets.

Figure 3 shows the training time of every model. On BDD and IDD, LoRA has trained the fastest on average, while Cityscapes, using no adapter, was the fastest on average. No adapter strategy trained faster than the other strategies for every run, when considering a single dataset. If we are to compute average training time across datasets, LoRA trained for 48 min., standard adapter for 63 min., and the no-adapter strategy for 50 min. on average. The numeric results used for creating Figures 2 and 3 are included in Appendix A.

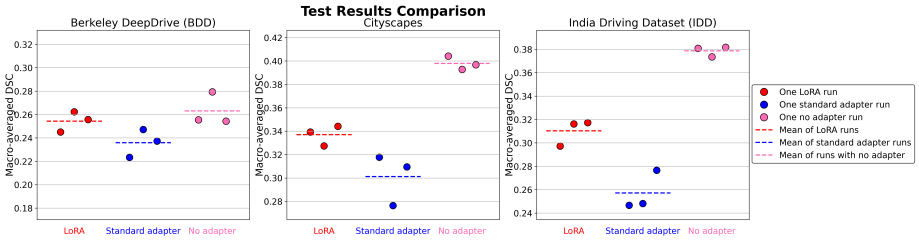


Figure 2: Macro-averaged DSC of every model, when evaluated on the test set.

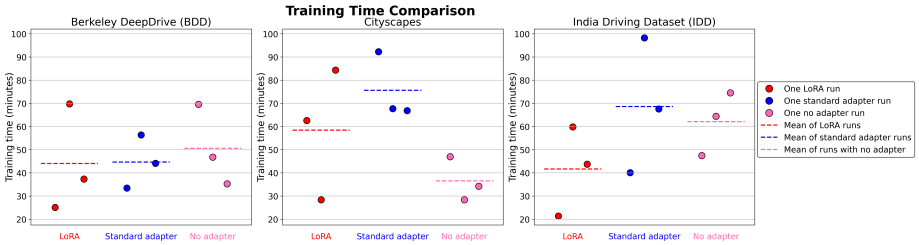


Figure 3: Training time of every model.

Figure 4 shows validation curves for all runs. It shows that the no-adapter strategy achieves the best model in terms of macro-averaged DSC on the validation set on all three datasets. It also shows that when excluding the no-adapter strategy, it is LoRA that achieves the best model in terms of macro-averaged DSC on the validation set on all three datasets.

Figure 5 shows the application of models from every adapter strategy to an arbitrary image from IDD, as well as the image itself and the corresponding ground truth. In every case, the model used was the highest scoring one according to Figure 2. We see that all the adapter approaches manage to predict the background and road. LoRA also finds the two motorcycles in the foreground as well as their riders. It also predicts the cars to the right as a mixture of car and truck, the person to the left as a rider, and it has incorrect artifacts, especially in the left part of the background. The standard adapter has a hard time predicting whether the riders in the foreground should be classified as riders or persons. It does, however, correctly predict the person to the left and the cars to the right, but mostly fails to correctly classify the foremost motorcycle correctly, and it classified the entire structure to the left as a traffic light, while also showing artifacts, especially at the border between the road and the background. The no-adapter strategy successfully segmented the riders and motorcycles in the foreground, the cars to the right, and the person to the left. It also manages to segment some smaller motorcycles and riders further down the road, approximately in the middle of the image.

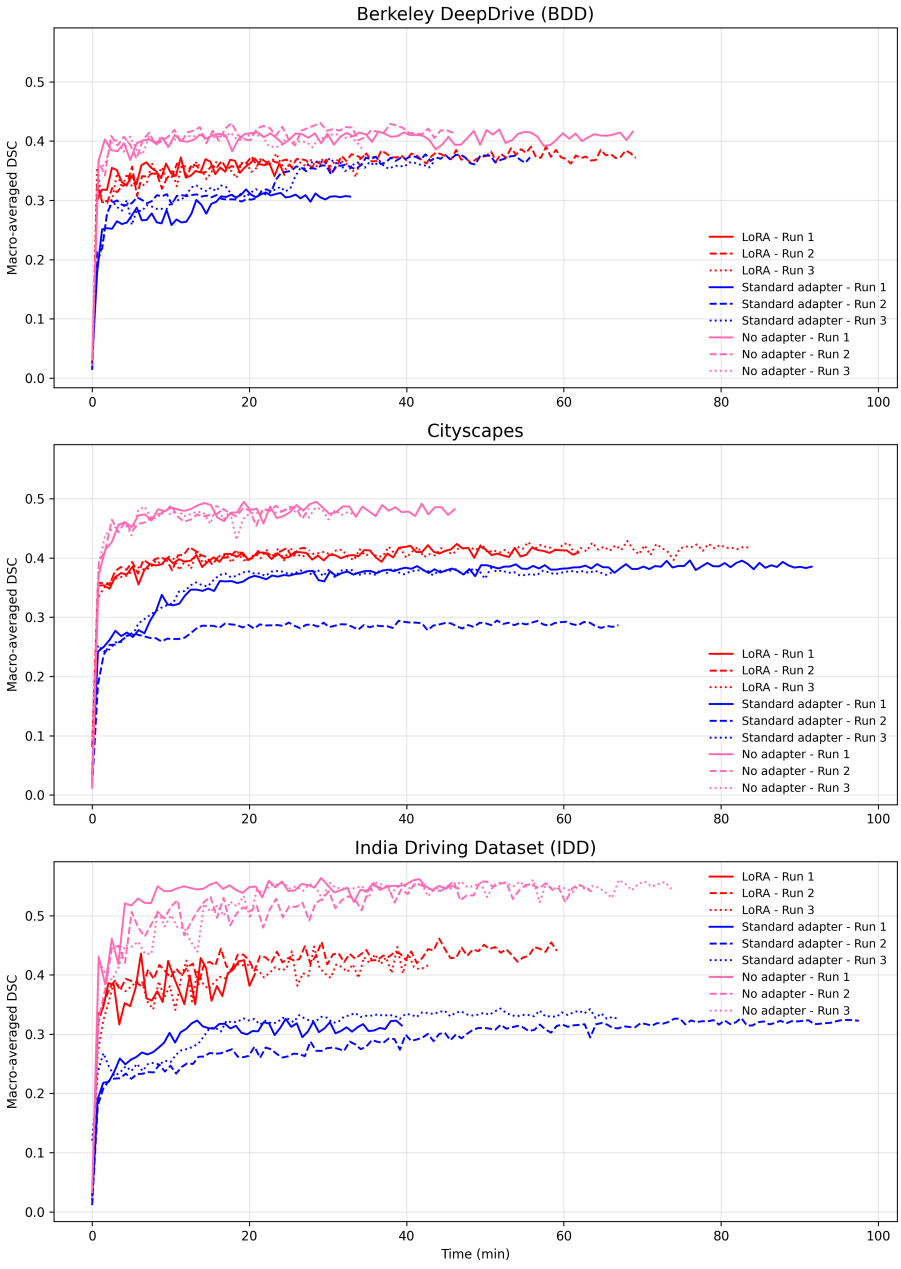


Figure 4: Macro-averaged DSC of every training run on the validation sets.

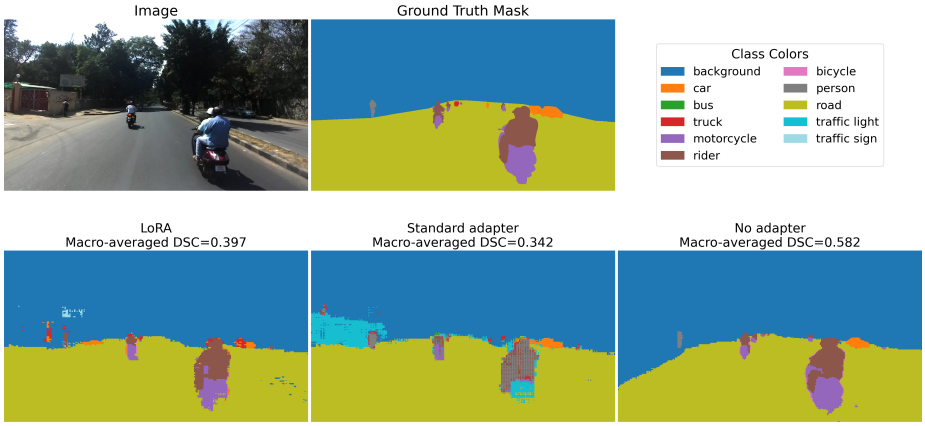


Figure 5: One arbitrarily chosen image from IDD, its ground truth mask, and the segmentations predicted by the highest-scoring model (according to Figure 2) for every respective adapter strategy on IDD.

4 Discussion

Firstly, it should be noted that with only three training runs for every configuration and no measure of statistical significance, the points made in the following should by no means be seen as definitive.

4.1 Interpretation of Results

For all datasets, the spread of macro-averaged DSC for the same adapter strategy is rather small, which indicates consistency and suggests, but doesn't confirm, that the observed results are not due to random chance.
That said, when interpreting Figure 2, the no-adapter strategy does appear to outperform the other strategies in terms of macro-averaged DSC rather consistently. For both the Cityscapes dataset and IDD, the gap from the no-adapter runs to LoRA seems rather well-defined, implying that if more repetitions of the experiments were conducted, substantially different results would need to be observed before the no-adapter strategy would not be performing better in terms of macro-averaged DSC. On BDD, the no-adapter strategy still achieves the highest macro-averaged DSC on average, but here it is rather close, and there is no well-defined gap separating it from the other approaches. For the BDD dataset alone, it would be difficult to argue with confidence that the no-adapter strategy is better, as it would be easier to imagine that more samples potentially could alter the situation.
That said, Figure 2 shouldn't stand alone, as there is no indication of the amount of computational resources used for training. Therefore, it should be interpreted in tandem with Figure 3. When analysing Figure 3, we see that, unlike in Figure 2, there appears to be a great spread among the samples; it is therefore difficult to conclude whether any adapter strategy will be faster than the others with the given experimental setup, if more samples were to be collected. Furthermore, it is interesting to note that, when keeping the dataset and adapter strategy

constant, large differences in training time do not seem to indicate large differences in macro-averaged DSC. That means the gain from training a model for a long time appears small. Considering the average training times as mentioned in Section 3 of 48, 63, and 50 min. for LoRA, the standard adapter and no adapter, respectively, LoRA and no adapter are very comparable, while the standard adapter took some more time. The longer training time means that the standard adapter has a slight unfair advantage when considering Figure 2 alone, but nevertheless, it still has the lowest average value of macro-averaged DSC for all datasets.

The evidence becomes even clearer when analysing Figure 4. Here we see that already after about five to ten min. of training, the no-adapter strategy has the highest macro-averaged DSC on the validation set, followed by LoRA. The separation mostly stays throughout training, although for BDD, the margins are smaller and the standard adapter even manages to have approximately the same macro-averaged DSC as LoRA for two out of three runs. Again, the most important thing to note here is how the no-adapter strategy has the highest macro-averaged DSC throughout almost the entire training. Although it should be kept in mind that these measurements come from the validation set, which was indirectly exposed during the construction of the experimental setup, before it was finalised, causing some degree of data leakage. It should also be noted that the macro-averaged DSC on the validation set for most runs quickly reaches a value close to the highest value it reaches during the entire training process, which further emphasizes the earlier point that longer training times do not have a large impact on the final macro-averaged DSC of the models.
For Figure 5 showing a qualitative result, specifically, the IDD dataset was chosen, as it was neither the best nor the worst dataset for all approaches according to the result in Figure 2, and therefore, it should hopefully show all models in an average setting. For the example image, it is hard to argue that the no-adapter strategy didn't do the best; it segments most classes well, although with some slightly inaccurate borders. LoRA appears to be second best, as it finds most objects rather well, although with some issues, and the standard adapter appears worst, as it finds most objects, but they're frequently mislabeled. The macro-averaged DSC for these predictions, as shown in Figure 5, does seem to agree with the qualitative interpretation of the predictions.

4.2 Implications of Results

The results imply that there is evidence, although not statistically significant, that not using an adapter achieves the highest macro-averaged DSC for the traffic domain for various geolocations when fine-tuning MobileSAM. That observation is unlike previous work conducted on the medical domain [3] by Gu et al, whose experiments show that the standard adapter was best by a small margin. One likely explanation for the discrepancy is the change of domain; Considering the large differences of the medical and traffic domains, it would be naive to assume that findings from one domain will seamlessly carry over to the other. Furthermore, this project did change the experimental setup (see Section 2.4), which is another potential source for discrepancies. The findings in Section 3 also conflict with the working hypothesis, which is that no strategy will achieve clearly better performance. Since the hypothesis is based on the

work of Gu et al [3], the above reasons discussing why the findings of this project differ from theirs also hold when explaining why the findings don't support the hypothesis.

The research question of this project was to investigate whether using an adapter for fine-tuning MobileSAM for the traffic domain for various geolocations is beneficial or not in terms of DSC. Based on the empirical experiments conducted, the results indicate that applying an adapter worsens the performance in terms of DSC. The lower DSC of the adapter-based approaches means that if one were to fine-tune a model for many geolocations with an adapter such that they only need to store one base model and one adapter for each geolocation, as mentioned in Section 1, they must accept lower performance in terms of DSC, compared to fine-tuning a model with no adapter for every geolocation of interest.
The findings of this project also demonstrate that adapting a base model without any adapters still has merit, even though the use of adapters has become very popular in recent times.

4.3 Limitations and Biases

The findings of this project should be seen in the light of the limitations present, which follow. Additionally, it should be noted that for most of the following points, such as preprocessing, a decision was made based on human judgment, which comes with associated biases.

4.3.1 Amount of Training Runs

Since every experiment was only conducted three times, the results will naturally carry an amount of uncertainty. It is possible that repeating the experiments many times could alter the findings.

4.3.2 Extrapolation to Higher-Performing Models

The models trained for this project will likely never find any real-world application, given that autonomous driving is a safety-critical system, and perception systems that don't exhibit high performance will likely not be used for autonomous driving. This means that for the findings of this project to find real-world application as is, one must assume that the findings still apply to higher-performing (whether that is in terms of DSC or other metrics assessing the quality of the segmentations) models, and it is currently unclear whether that assumption is true.

4.3.3 Experimental Setup and Hyperparameter Choices

All experiments were conducted with the same experimental setup and hyperparameter choices. It is unclear whether the findings will hold when these are changed.

4.3.4 Choice of Adapters

Only two types of adapters were used in this project. It is possible that other adapters would achieve different results.

4.3.5 Datasets and Geolocations

Only three datasets and geolocations were used, and it is unclear whether the finding will generalise to other datasets or geolocations.

4.3.6 Metrics

This project builds on the assumption that macro-averaged DSC is an appropriate way to evaluate the quality of segmentations. Measuring the quality of the segmentations with different metrics might yield different results.

4.3.7 Training Time as a Proxy for Computational Cost

It was assumed that training time was a reasonable proxy for computational cost, which may not be true.

4.3.8 Preprocessing

It is not guaranteed that the results will generalise for different preprocessing approaches. Especially interesting is whether the semantic subset selection could benefit some adapter strategies more than others.

4.4 Future Research

As this project is subject to some major limitations (see Section 4.3), it would make sense for future research to address these first. The limitation of the low amount of training runs could naturally be addressed by increasing the number of training runs. The limitations regarding experimental setup, hyperparameter choices, preprocessing, datasets, and geolocations could all be addressed by conducting new experiments while varying these elements. Furthermore, one could investigate other metrics to evaluate whether they're more appropriate, and compute flops during training instead of training time. Lastly, one can train higher-performing models and evaluate whether the findings of this project still apply, though doing that in isolation could be challenging, as it would likely require changing the experimental setup or hyperparameters.

5 Conclusion

When fine-tuning a base model like MobileSAM for traffic segmentation for specific geolocations, it has previously not been clear whether applying an adapter is beneficial in terms of DSC.
This project has investigated this by empirically evaluating three fine-tuning strategies: LoRA, standard adapter blocks, and no adapter for fine-tuning MobileSAM to three different geolocations. Upon analysing the results, it can be concluded that there is evidence implying that it is beneficial not to use an adapter. Although not statistically significant, the findings are rather clear, as the no-adapter strategy has outperformed the adapter approaches by a sizeable margin on two datasets, while still scoring higher on average on the third dataset.

References

[1] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350. URL: https://arxiv.org/abs/1604.01685.

[2] Hanxue Gu et al. *finetune-SAM: Fine-tuning Segment Anything Model for Medical Image Segmentation*. https://github.com/mazurowski-lab/finetune-SAM. Accessed: 2025-10-26. 2024.

[3] Hanxue Gu et al. "How to build the best medical image segmentation algorithm using foundation models: a comprehensive empirical study with Segment Anything Model". In: *Machine Learning for Biomedical Imaging* 3 (2025). Accepted for publication at the Journal of Machine Learning for Biomedical Imaging (MELBA), 86a6. DOI: 10.59275/j.melba.2025-86a6. arXiv: 2404.09957 [cs.CV]. URL: https://arxiv.org/abs/2404.09957.

[4] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 2790–2799. arXiv: 1902.00751. URL: https://arxiv.org/abs/1902.00751.

[5] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *arXiv preprint arXiv:2106.09685* (2021). arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[6] Tarun Kalluri, Wangdong Xu, and Manmohan Chandraker. "GeoNet: Benchmarking Unsupervised Adaptation Across Geographies". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Project page: https://tarun005.github.io/GeoNet/. 2023, pp. 15368–15379. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Kalluri_GeoNet_Benchmarking_Unsupervised_Adaptation_Across_Geographies_CVPR_2023_paper.html.

[7] Wayve Technologies Ltd. *Crossing the Pond and Beyond: Generalizable AI Driving for Global Deployment*. https://wayve.ai/thinking/multi-country-generalization/. Accessed: 2025-10-26. Mar. 2025.

[8] Girish Varma et al. "IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments". In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019, pp. 1371–1380. DOI: 10.1109/WACV.2019.00147. URL: https://arxiv.org/abs/1811.10200.

[9] Fisher Yu et al. "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020. arXiv: 1805.04687 [cs.CV]. URL: https://arxiv.org/abs/1805.04687.

[10] Chaoning Zhang et al. "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications". In: *arXiv preprint arXiv:2306.14289* (2023). arXiv: 2306.14289. URL: https://arxiv.org/abs/2306.14289.

Appendix

A Raw Test Results & Training Time Data

All values are rounded to three decimals for the macro-averaged DSC table and to the nearest minute for the training time table.

Macro-averaged DSC for every run	BDD	Cityscapes	IDD
LoRA	0.245	0.339	0.297
	0.262	0.327	0.316
	0.256	0.344	0.317
Standard adapter	0.223	0.318	0.247
	0.247	0.276	0.248
	0.237	0.310	0.277
No adapter	0.256	0.404	0.381
	0.279	0.393	0.374
	0.254	0.397	0.382

Training time for every run (min)	BDD	Cityscapes	IDD
LoRA	25	63	21
	70	28	60
	37	84	44
Standard adapter	33	92	40
	56	68	98
	44	67	68
No adapter	70	47	47
	47	28	64
	35	34	75

B AI-declaration

UCPH's AI declaration

Declaration of using generative AI tools
☒ **I/we have used generative AI as an aid/tool** *(please tick)* ☐ **I/we have <u>NOT</u> used generative AI as an aid/tool** *(please tick)* *If generative AI is permitted in the exam, but you haven't used it in your exam paper, you just need to tick the box stating that you have not used GAI. You don't have to fill in the rest.*
List which GAI tools you have used and include the link to the platform (if possible): *ChatGPT: [https://chatgpt.com/]*
Describe how generative AI has been used in the exam paper: *1) Purpose (what did you use the tool for?)* *For generating ideas and structures for code* *2) Work phase (when in the process did you use GAI?)* *For all phases that required coding* *3) What did you do with the output? (including any editing of or continued work on the output)* *I modified the output to fit my exact needs* *Please note:* Content generated by GAI that is used as a source in the paper requires correct use of quotation marks and source referencing. Read the guidelines from Copenhagen University Library at KUnet here.