Stream-based Active Learning by Exploiting Temporal Properties in Perception with Temporal Predicted Loss - Supplementary Material

Sebastian Schmidt[1,2]
sebastian95.schmidt@tum.de
Stephan Günnemann[2]
s.guennemann@tum.de

[1] BMW Group
Munich, Germany
[2] Technical University of Munich
Data Analytics and
Machine Learning Group,
Munich, Germany

A The A2D2 Streets Dataset

We created a scene classification dataset for an operational domain detection task. This task is essential for autonomous vehicles as it reflects whether they can operate safely in this environment. For example, construction sites are a domain where more caution is required. For the dataset, we used the image data of the A2D2 [5] which provides temporally coherent frames structured in different drives. We assigned the classification labels urban, highway, country road and construction site to the images describing the current driving environment. The dataset contains several recorded drives in southern Germany, with around 680 frames on average per recording. The frames are timestamped with a high frequency of up to 10 Hz so that the temporal change of the samples can be evaluated meaningfully. Although the rate is not constant due to sensor synchronization, the optical flow remains stable and does not get lost. The temporal coherence with a high frequency of sampled images brings the risk of selecting redundant samples in a batch. Due to the nature of the drives, the latent space representation is naturally clustered in the specific drives shown in Figure 2.

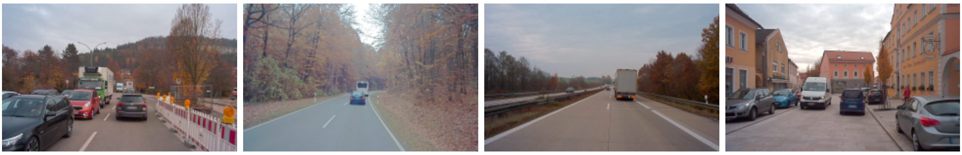

(a) Construction site (b) Country road (c) Highway (d) Urban

Figure 1: Overview of the different classes in the A2D2 streets dataset.

The recorded drives are not split and divided as a whole recording into an initial labeled pool and unlabeled pool for training as well as validation and test set as shown in Table 1. In

Assignment	Sessions		
initial labeled	20181107_132730	20181108_091945	
unlabeled pool	20181107_133258	20181108_084007	20180807_145028
	20180810_142822	20180925_135056	20181008_095521
	20181107_132300	20181204_154421	20181204_170238
validation set	20180925_101535	20181016_125231	20181204_135952
test set	20180925_124435	20181108_123750	20181108_103155

Table 1: This table shows the dataset split into internal labeled and unlabeled pool training set as well as validation and test set for A2D2s.

the stream-based setups, the unlabeled drives are fed as streams into the AL algorithm. The images have been resized to 120 × 72 pixels. The training dataset has been shuffled.

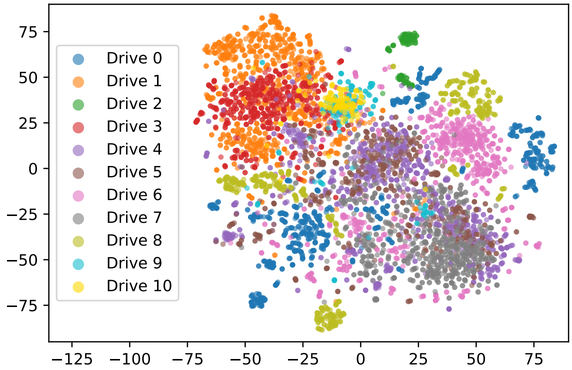


Figure 2: t-SNE analysis with perplexity 30 of the different recorded drives from the training set in A2D2s.

B The GTA V Streets Dataset

We created the GTA V streets dataset [1] as the first classification dataset designed for active learning, having temporal coherence. Like the A2D2 streets dataset, we designed an operation domain detection task for our dataset and added the labels for highway, urban, country road and gravel road. However, the dataset was recorded from a game, not the real world like A2D2, so the environment is more manageable and potential variables like weather can be avoided.

Figure 3 shows examples of the different classes. The dataset contains seven recordings with dedicated routes that do not intersect each other to avoid overlaps between the training, validation and test split. Figure 4 shows a map of the different routes. All routes are recorded during the day hours and similar weather conditions. Additionally, we designed the routes such that all classes are present in each route. However, the share of classes in each route varies. By sampling the routes at 10Hz, the dataset contains around 35000 frames. Each

---

[1] `https://www.cs.cit.tum.de/daml/tpl/`

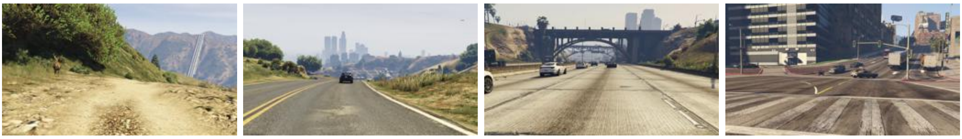

(a) Gravel Road (b) Country road (c) Highway (d) Urban

Figure 3: Overview of the different classes in the GTAV streets dataset.

Assignment	Sessions
initial labeled pool	Route6
unlabeled pool	Route2, Route4, Route5, Route7
validation set	Route1
test set	Route3

Table 2: This table shows the dataset split into internal labeled and unlabeled pool training set as well as validation and test set for GTAVs.

frame has a size of 128 × 72 pixels. The dataset is split into an initial labeled pool, an unlabeled pool, and a validation and test set for the experiments. The exact split is shown in Table 2.

C Experiment Details

In this section, we highlight our experiment details and hyperparameters. As the datasets are highly redundant due to the recording character, we applied early stopping on the validation accuracy, setting the parameter for patience to 30. We followed the official implementations of Resnet18 [6] and VGG11 [9] provided by PyTorch [7], except for minor modifications. For Resnet18, we added two fully connected layers with dropout layers between them to the head. For VGG11, we reduced the hidden layer size to 1024 and 512. Additionally, the convolutional layers were initialized using the pre-trained ImageNet [4] weights provided by PyTorch. For the ResNet18 model, we attached the loss learning modules after each of the four blocks. In the case of VGG11, we attached the loss modules after the last four max pooling layers.

For GTAVs, we set the batch size to 128 and used a learning rate of 0.001 with SGD with a momentum of 0.9. For A2D2s, we used a batch size of 64 with a reduced momentum of 0.8. As the redundancy of the dataset is quite high, we chose an early stopping strategy on the validation accuracy with patience 30. We used the checkpoint with the highest validation accuracy to continue. For the loss learning module, we split the early stopping to a first, detached the gradients from the loss module to the task model and a second early stopping to stop the training. Different detachment points are examined in Section E.2. We set the margin ζ and ξ to 0.5. The scale of the $\mathrm{L1}_{\text{margin}}$ loss λ is set to 0.5. The scaling factor η is set to 1 for the combined loss.

Further, we neglected extensive augmentations for our active learning setup. As we are using a stream-based setup, using the normalization to zero mean and unit standard deviation of the whole training pool is not possible. As some changes between the drives can occur, we used histogram equalization as a data preprocessing step instead. Due to the high redundancy of the dataset, we only select between 15% to 42% of the training data. All experiments

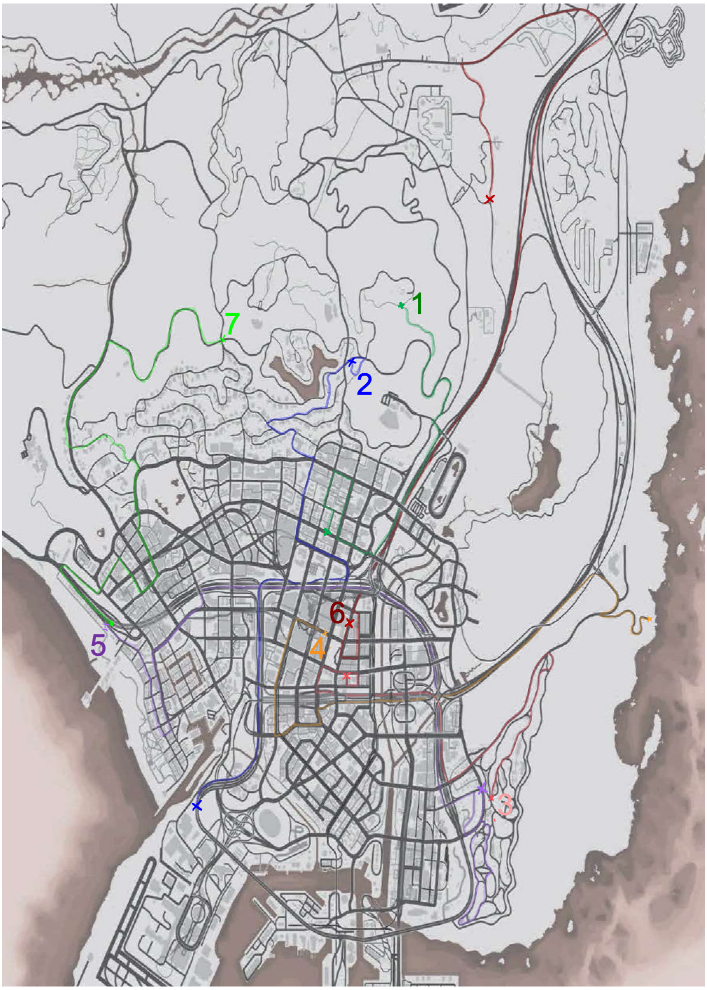


Figure 4: The routes of the GTAVs dataset, best seen in color.

are conducted three times with the seeds 1,42,64. If not named differently, the parameters suggested by the authors are taken for the state-of-the-art methods. The number of forward passes for MC dropout is set to 10.

The hyperparameters describe the weights in a combined loss function and are defined with a hyperparameter search. As the loss is a combined loss, the loss learning task influences the weights of the layers used for the perception task of the primary model. Therefore, they should be chosen such that strong regularizing and disturbance effects, leading to decreased task performance, are avoided. Too strong criteria on loss learning loss and a bad detachment point lead to underperformance in task results. The parameters can be appropriately selected by comparing a model with a loss module and a model without a loss module in the initial training. The second goal is tuning the parameters based on the ranking quality, whereas the first goal is more important, which can be achieved by a rough grid search. Our findings show that the approach is robust and not too sensitive to small changes.

Class Index	Classes
Ignored	Rain dirt, Blurred area
Nature	Nature object
Buildings	Buildings
Traffic Guide	Electronic traffic, Irrelevant Signs, Traffic guide obj. Signal corpus, Poles, Grid structure Traffic signal 1 Traffic signal 2, Traffic signal 3 Traffic sign 1, Traffic sign 2, Traffic sign 3
Non Drivable	Non-drivable street, Parking area
Ego Car	Ego car
Bicycle	Bicycle 1, Bicycle 2, Bicycle 3, Bicycle 4
Pedestrian	Pedestrian 1, Pedestrian 2, Pedestrian 3
Small Moving Objects	Small vehicles 1, Small vehicles 2, Small vehicles 3
Moving Medium Objects	Car 1, Car 2, Car 3, Car 4
Moving Big Objects	Truck 1, Truck 2, Truck 3 Utility vehicle 1, Utility vehicle 2, Tractor
Sky	Sky
Street Areas	Speed Bumper, Driveable cobblestone Slow drive area, RD normal street
Guiding	Road blocks
Lane Markings	Zebra crossing, RD restricted area, Painted driv. instr.
Lines	Solid line, Dashed line
Sidewalks	Sidebars, Curbstone, Sidewalk
Obstacles	Obstacles / Trash, Animals

Table 3: Class setting for semantic segmentation.

D Semantic Segmentation Experiments

This section highlights the exact settings of our semantic segmentation experiments based on object detection subset from A2D2 using a DeeplabV3 Chen *et al.* [2] model. The loss learning layers are attached in the same way to the ResNet34 backbone as for ResNet18. Additionally, we added loss learning modules to each Atrous Spatial Pyramid Pooling (ASPP) block of the DeeplabV3 module. We used a version of the dataset also containing the bounding box labels. The splits and query size are set identically to the parameter in the classification experiments. The same holds for TPL and loss learning parameters, which are taken out of the box from the classification experiments. For the backbone, pre-trained weights are used. For the training, we used a batch size of 32 and a learning rate of 0.05 for the head and 0.005 for the backbone. We did not use any augmentation but the same preprocessing as used in the classification experiments, except for the image size. The images have been resized to 640x400. We compared our TPL method to loss learning, a random selection and a MC dropout entropy-based selection using the mean of the pixel entropy values to determine the value of each sample. As the exact class remapping used in [5] is not given, we defined 17 classes and provided the mapping in Table 3. The classes "ignored" and "ego_vehicle" are excluded from the evaluation.

Method	Cycle 1	Cycle 2	Cycle 3	Cycle 4
TPL retrain	0.802(8)	0.923(3)	0.962(5)	0.967(2)
Loss learning retrain	0.827(11)	0.878(8)	0.937(11)	0.938(6)
Random retrain	0.839(18)	0.918(5)	0.950(9)	0.956(5)
TPL continuous	0.833(11)	0.922(6)	0.950(8)	0.960(11)
Loss learning continuous	0.813(17)	0.879(19)	0.938(12)	0.940(3)
Random continuous	0.844(4)	0.902(14)	0.942(8)	0.949(6)

Table 4: Different training strategies for active learning.

E Ablation Study

Additionally to the main experiments, we evaluated the training strategy in Section E.1, the detachment of the loss learning module in Section E.2 and our modifications to the loss learning module loss in Section E.3.

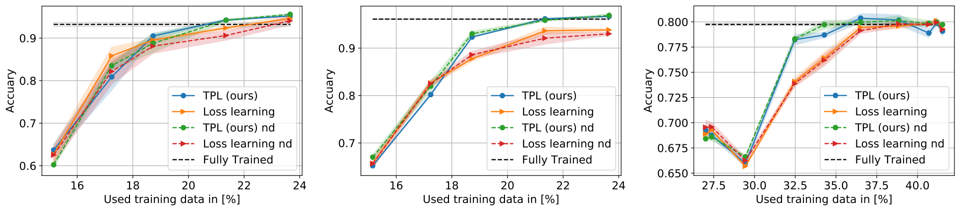
E.1 Training Strategy

The topic of the training scheme of AL cycles is relatively unexplored. The model can be trained from scratch or the current model state can be enhanced and reused. As most works tend to retrain the model, we use this strategy. Additionally, we can avoid side effects due to continuous training strategies and focus on the selected properties in this way. As [8] and [3] reported interesting results by using different strategies, we conducted experiments to evaluate the decision of [11] to use a continuous training strategy. The results are reported in Table 4.

As can be seen, the continuous training strategy has a minor effect on all methods evaluated. While TPL and random are decreased at the last cycles and boosted at the first cycles, it is the other way around for loss learning. The inconsistency of the results shows how difficult an interpretation and a distinction between the selection method and training strategy is. Interestingly, the order of the methods does not change. Due to the small effect on performance, continuous training strategies are mainly suitable for time savings.

E.2 Loss Learning Module Detachment

In this section, we evaluate the point of detaching the gradients from the loss learning module. Fixed epochs, as proposed by [11], are suboptimal for datasets with lower diversity as they tend to overfit. As described in Section C, we follow an early stopping strategy for the detachment. To justify our proposed approach, we compared the early stopping detaching of gradients with the approach of not detaching at all and depicted the results in Figrue 5. The figures indicate that the performance of TPL is almost independent of the detachment point for the GTAVs dataset in the areas of higher-used training data. While at the first two phases, the accuracy of TPL differs. This can also be observed in the loss learning approach. However, the effect is more present for the region between the minimum and maximum data selection size. For A2D2s in Figure 5(c), the effect on loss learning is neglectable, while TPL shows in the region of the fully trained network differences and achieves higher saving when not detached. For A2D2s, it should be considered that due to the independent recordings in the training, validation and test set, the distributions can have limitations in overlap. Therefore, the early stopping on the validation set has limitations.



(a) GTAVs ResNet18 (b) GTAVs VGG11 (c) A2D2s VGG11

Figure 5: Detachment comparison, non-detached is abbreviated with "nd".

Concluding the experiment shows that both methods are rather unsusceptible for the detachment point. It can be considered a hyperparameter for minor optimization purposes. The finding additionally underlines the downsides of learning-based approaches like VAAL [10], loss learning [11] and CoreGCN [1], which showed increased data saving potential by the cost of additional models and hyperparameters.

E.3 Loss Learning Loss Modifications

In this section, we evaluate our modification to the loss learning loss and highlight the influence of different parameters.

L1 regularization:

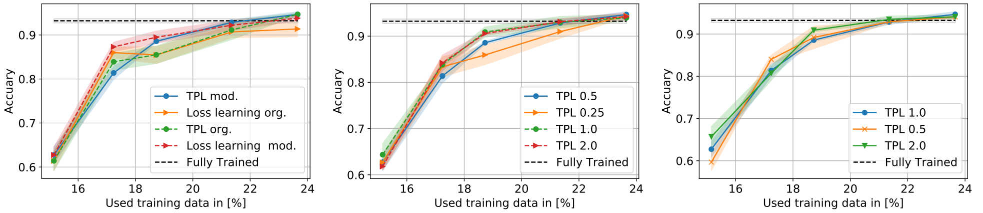
In order to evaluate the effect of our L1 regularization adaption to the loss learning module loss function in Equation 3, we compare TPL and the vanilla loss learning with both losses in Figure 6(a). The plot shows that our loss function adaption, as well as our Temporal predictive loss function, both improve the vanilla approach individually. As the combined approach delivers the best performance, both effects are cumulative.

Margin selection:

In the following study, we highlight the effect of the Xi and Zeta margin in Equation 3. For simplification, we use for ζ and ξ the same value and show their effects in Figure 6(b). As the figure indicates, the margin factors have a minor role compared to the loss function. While higher margins improve the performance in low-data regions, our selected value shows the highest accuracy. A more fine-grained parameter search would potentially improve the results further. Conversely, the limited effect of tuning this parameter indicates that a low-effort parameter search is sufficient, making our method easier to apply.

Lamda factor selection:

Lastly, we examine the lambda factor for the regularization of the loss learning module loss. Figure 6(c) shows different values for this factor and indicates that the lambda factor has an impact on regions of low data, which is reduced for regions with more data. While higher regularization improves the performance in low-data regions, lower and higher regularization improves the result for low-data regions. The differences between different lambda values are strongly decreasing for regions with more data. We choose based on the highest accuracy the value one, which has also been chosen by Yoo *et al*. [11]. Also, in this case, a more fine-grained parameter search has the potential to improve the results further. However, the limited effect of tuning this parameter shows that a low-effort parameter search is sufficient, proving again that our method is easy to apply.



(a) Loss function modification with vanilla comparison for ResNet18 on GTAVS.

(b) Loss function margin values ξ and ζ study for ResNet18 on GTAVS.

(c) Lambda factor comparison for ResNet18 on GTAVS.

Figure 6: Ablation study of different parameters of the loss learning loss function.

References

[1] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9583–9592, 2021.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[3] Feras Dayoub, Niko Sunderhauf, and Peter I. Corke. Episode-based active learning with bayesian neural networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, volume 2017-July, pages 498–500, 2017. ISBN 9781538607336. doi: 10.1109/CVPRW.2017.75.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 248–255, 2009.

[5] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset. *arXiv*, 2004.06320, 2020. URL `https://www.a2d2.audi`.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach,

H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *In Proceedings of the International Conference on Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019.

[8] Sebastian Schmidt, Qing Rao, Julian Tatsch, and Alois Knoll. Advanced active learning strategies for object detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 871–876, 2020.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[10] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3 2019.

[11] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.