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

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

Abstract

Active learning (AL) reduces the amount of labeled data needed to train a machine learning model by intelligently choosing which instances to label. Classic pool-based AL requires all data to be present in a datacenter, which can be challenging with the increasing amounts of data needed in deep learning. However, AL on mobile devices and robots, like autonomous cars, can filter the data from perception sensor streams before reaching the datacenter. We exploited the temporal properties for such image streams in our work and proposed the novel temporal predicted loss (TPL)[1] method. To evaluate the stream-based setting properly, we introduced the GTA V streets and the A2D2 streets dataset and made both publicly available. Our experiments showed that our approach significantly improves the diversity of the selection while being an uncertainty-based method. As pool-based approaches are more common in perception applications, we derived a concept for comparing pool-based and stream-based AL, where TPL outperformed state-of-the-art pool- or stream-based approaches for different models. TPL demonstrated a gain of 2.5 percent points (pp) less required data while being significantly faster than pool-based methods.

1 Introduction

To apply neural networks in real-world situations, vast amounts of data must be labeled. Active learning (AL) is a technique to minimize the labeling effort by letting a machine learning model choose the data to be labeled by itself. The two main scenarios are pool-based and stream-based AL [54]. The most common pool-based AL scenario, sketched in Figure 2(a), is a cyclic process of selecting batches $\mathcal{B}$ of the most promising samples based on a query function from a pool of data stored on a datacenter. The model is retrained after the selection to start the next iteration of the AL cycle. In contrast, stream-based AL, depicted in Figure 2(b), assumes an inflow of samples as a stream and the model decides if a sample should be saved and labeled or disposed. As the data is examined only once for a decision,

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

the data is not required to be present in a datacenter. In classic stream-based AL, the model is updated after each selection [34]. In these scenarios, different selection methods, also called
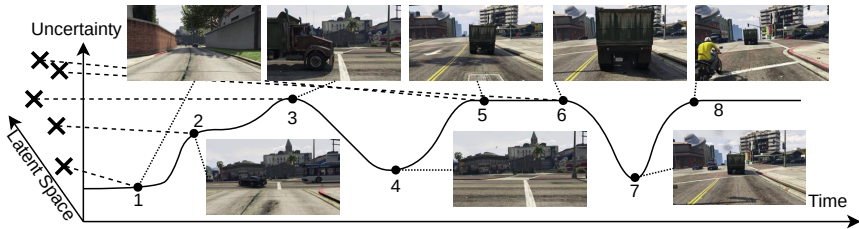

Figure 1: GTAVs uncertainty flow - temporal coherent latent space and uncertainty for TPL.

querying strategies, are applied to select the data for labeling. The three main categories are uncertainty-based, diversity-based and learning-based AL [29]. Uncertainty-based AL methods predict or approximate an uncertainty value for each sample [7, 14]. Diversity-based methods select samples based on the latent space coverage of the dataset. Usually, they perform an optimization, which requires constant access to the labeled and unlabeled dataset. Learning-based approaches train an additional model to either predict a pseudo-uncertainty value or decide if a sample should be selected directly. The usage of unlabeled data for training the additional model or optimization limits diversity- and learning-based methods primarily to pool-based scenarios.

Real-world perception tasks, like autonomous driving, robotic perception and environmental sensing, are extraordinarily data intensive. AL is an essential research topic to handle this amount of data for tasks like classification, object detection and semantic segmentation. However, researchers in this area, particularly autonomous driving, are focused on pool-based AL [11, 12, 18]. Since the data is not generated in the datacenter, data logistics and preparation limit the possibilities to apply and scale AL approaches to open-world perception problems requiring a lot of data. Divergent from recent efforts, we examine stream-based AL to use these computations' advantages and make AL scaleable for the complexity of real-world applications. By running stream-based AL directly on mobile devices used in these applications, data can be collected through agents without transferring it to a datacenter.

To incorporate the real-world challenges, we propose to benchmark AL on camera or sensor stream data directly instead of using the datasets designed for various perception tasks. We apply AL directly on temporally coherent camera streams to enable agent-wise processing, which reduces preprocessing efforts. The temporal coherence of the sensor streams enables the representation of the uncertainty as a continuous function of time, depicted in Figure 1. We exploit the temporal coherence and examine the changes in uncertainty to significantly increase the diversity of the selected samples without using optimization techniques requiring access to unlabeled data or extensive optimizations.

Contributions: (1) We introduce a new perspective on AL in perception and use it to derive our novel Temporal Predicted Loss (TPL) AL approach. By incorporating temporal information into uncertainty estimation, TPL increases the diversity of the batch selection. (2) We introduce two operational domain detection datasets, the GTA V streets (GTAVs) based on game recordings and the Audi Autonomous Driving Dataset streets (A2D2s) derived from the A2D2 [16] with temporal coherence between the samples. (3) We define a comparison approach between pool-based and stream-based AL and use it to compare TPL with state-of-the-art approaches, showing an improvement of 2.5 pp less required data.
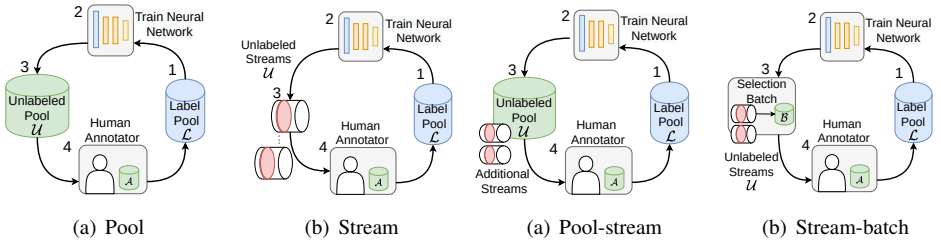
Figure 2: Classic active learning scenarios.   Figure 3: Derived active learning scenarios.

2   Related Work

While many authors did significant research on pool-based AL, stream-based AL has lost attention with the rise of deep learning. However, stream-based AL is crucial since the amount of vision data that can be uploaded for pool-based AL is limited.

In pool-based AL Sener and Savarese [32] defined AL as a core-set selection problem, creating a diversity-based AL approach, denoted as CoreSet. A K-center approach selects samples, minimizing the maximum distance to not-selected ones. Ash *et al.* [1] combined the distance representation of the latent space with pseudo labels based on the highest one-hot encoded values to generate gradients to include prediction uncertainty. Uncertainty-based approaches are mostly sampling-based. Monte Carlo (MC) dropout [14] uses multiple forward passes with active dropout layers during the prediction to create a distribution over the class predictions, Beluch *et al.* [2] used ensembles to create a distribution. The uncertainty is estimated by either the entropy or the mutual information of the distribution. Houlsby *et al.* [19] introduced this as Bayesian active learning by disagreement (Bald). BatchBald [23] takes the diversity of the selected batch into account by calculating the joint mutual information to reduce redundant information in a batch. Loss learning [37] is a learning-based approach estimating a pseudo-uncertainty with only one forward pass. A loss learning module is added to specific layers of the prediction network to predict the network's loss, which is used as pseudo-uncertainty for sample selection. Novel learning-based methods like variational adversarial active learning (VAAL) [36] use unlabeled data, too. An autoencoder is trained to learn a latent space representation of the data based on the labeled and unlabeled sets. A discriminator model is trained to discriminate between labeled and unlabeled data based on latent space encoding. Kim *et al.* [22] extended VAAL [36] by adding a loss prediction module and included the predicted loss to the latent space. Caramalau *et al.* [5] introduced a more diversity-oriented learning-based approach using unlabeled data with a sequential graph convolutional network (CoreGCN). The authors used the distance between the features of the task model to calculate an adjacency matrix for a graph containing labeled and unlabeled data. A graph neural network predicts the nodes' value for being labeled.

Stream-based AL has a different focus than pool-based AL. An important topic of research here is conceptual drifts, where the underlying distribution changes over time [27, 38]. In stream-based AL, the selection is often described as a submodular optimization problem where the value of an added labeled sample is dependent on the labels already present. As solving these problems is computationally expensive, stream-based greedy algorithms are an important field of research [13]. One approach designed for data summarization is Sieve-Streaming++ [21]. However, stream-based AL has rarely been used for perception tasks,

especially when using deep learning models. It is more present in low dimensional data streams like chemical plant data [4]. In perception, Mondrian forests have been used in combination with online learning Narr *et al*. [25]. For non-deep neural network models, online and incremental learning [7] are often combined with AL. For neural networks, Puck *et al*. [28] used different uncertainty-based approaches on a mobile robot that stored selected data during operation. Senzaki and Hamelain [33] used the Sieve-Streaming++ [21] and applied it to the models' latent space for AL on edge devices (ALED) but neglected temporal and stream properties. We consider online learning and conceptual drifts as orthogonal problems to this work and focus on connecting pool- and stream-based AL for perception.

AL on temporal data has only been used for pool-based AL in previous works. Bengar *et al*. [3] used object detection metrics requiring ground truth to build a temporal graph and select samples with energy minimization. For this purpose, the authors created the SYNTHIA-AL dataset with short snippets and a high sampling rate based on the SYN-THIA [30] dataset. Schmidt *et al*. [31] used the object detection classification uncertainty estimated by the entropy over a time horizon in a pool-based scenario. Huang *et al*. [20] used temporal information to avoid multiple MC dropout forward passes [14] for semantic segmentation by combining the uncertainty prediction of a single forward pass with a flow network to calculate the uncertainty as a moving average over a time horizon.

Although many topics have been covered, particularly in pool-based AL, stream-based AL approaches in perception are rare. The most recent approaches from pool-based AL are diversity-based and learning-based, which are unsuitable for stream-based AL due to the required access to unlabeled data. Only the uncertainty-based approaches and the loss learning [37] approach are applicable for stream-based AL. Temporal properties have only been used to save the computation of multiple forward passes. In stream-based AL, temporal properties remain a relatively unexplored research topic.

In contrast to previous works, we leverage temporal coherence properties in a stream-based AL setup and improve the selection process in terms of speed and performance. As this research direction is unexplored yet we only consider uncertainty-based query methods as supportive for our work.

3 Temporal Information in Active Learning

Most perception datasets do not contain temporal data. The commonly used datasets Kitti [15] and Cityscapes [8] aim to have a good diversity to be highly generalizable and do not provide temporally ordered data streams. The same applies to classification datasets like Cifar10 [24], often used to benchmark AL. Since these datasets are created for classification, object detection, or semantic segmentation tasks, temporal coherence information is of less importance. Benchmarking AL on these datasets is sub-optimal and only shows potential label savings on datasets that have been designed for diversity. Instead, we propose to evaluate AL on camera or sensor streams directly to incorporate real-world challenges like the high correlation between successive samples and avoid additional manual work besides labeling.

Thus, we introduce the A2D2s data and the GTAVs dataset as two benchmark datasets[2]. These classification datasets feature operational domain detection with temporally coherent frames structured in different recorded drives. Operational domain detection is a critical task in autonomous driving and mobile robotics to estimate if the current state of the environment

---

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

is safe to operate. With the A2D2s we extend the A2D2 [16] images by operational domain labels. To create the GTAVs dataset, we recorded and labeled drives based on the game Grand Theft Auto V (GTA V). We designed dedicated routes for the individual drives to ensure they overlap free. An overview map is given in Appendix B. Each dataset has four different classes describing the environment. In A2D2 are urban, highway, country road and construction site. For GTAVs, the classes are urban, highway, country road and gravel road. Further details can be found in Appendix A and B.

As elaborated in Section 2, pool-based and stream-based AL are pretty detached. To enable comparisons of methods for the different scenarios, we modified the original AL cycles from Figure 2 to create the intermediate scenarios pool-stream and stream-batch depicted in Figure 3. However, a stream-batch setup was already implicitly used by [28, 33]. Both scenarios reflect growing datasets closer to mobile robotic and autonomous driving applications. In detail, they start with a small labeled dataset (1), which is used to train a model (2). The pool-stream scenario depicted in Figure 3(a), reflects a continuous data collection with an unlabeled pool (3) growing at every cycle. All samples, including the unlabeled ones, can be seen and used multiple times. To reflect limited recording and transferring capabilities, we change the scenario to a stream-based one. We remove the pool to create the stream-batch scenario depicted in Figure 3(b), where the current stream is visible for the model only once. In contrast to the classical stream scenario, a batch $\mathcal{B}$ (3) of a maximum size of b can be selected from a stream in each cycle.

4 Temporal Predicted Loss - Exploiting Temporal Information in Active Learning

For stream-based AL, uncertainty-based approaches remain suitable for choosing a labeling batch $\mathcal{B}$. The methods select the samples with the highest predictive uncertainty estimate σ_p. However, these methods cannot avoid overlapping information in the selected batch.

TPL exploits the relation between consecutive uncertainty values to increase the diversity of the selected batch. By defining the drives of the dataset as consecutively ordered streams, the predictive uncertainty σ_p and the stream samples x can be represented as a function of time $\sigma_p(t)$ and $x(t)$ respectively. Due to the temporal coherence properties of a sensor stream and assuming real-world situations (or their simulations), the change between two samples is continuous. In situations unknown to the model, the slope of uncertainty exceeds the steady behavior of known situations. By considering the temporal change of the uncertainty, we can detect these situations and filter similar samples with a close temporal relation and similar uncertainty values. Figure 1 depicts an exemplar continuous uncertainty function for a scene flow of GTAVs route 1 and shows the sudden change due to an object appearing (3) and the plateau of closely related scenes (5) and (6).

We use a loss learning module f_σ [37] and its predicted loss as (pseudo-)uncertainty $\tilde{\sigma}_x$ to estimate σ_p. The sampling of uncertainty-based approaches is problematic for stream-based applications due to increased computational costs. The loss module comprises a global average pool layer, a fully connected layer, and a ReLU activation function. Several modules are attached to specific network layers and concatenated to predict the loss of a given sample x. The attachment points are described in Section 5. The predicted loss $\tilde{\sigma}_x$ of a sample x is given in Equation 1. The loss module and latent space representations depend on the currently selected training set $\mathcal{D}_i$ and are updated after each AL cycle i. For readability, we

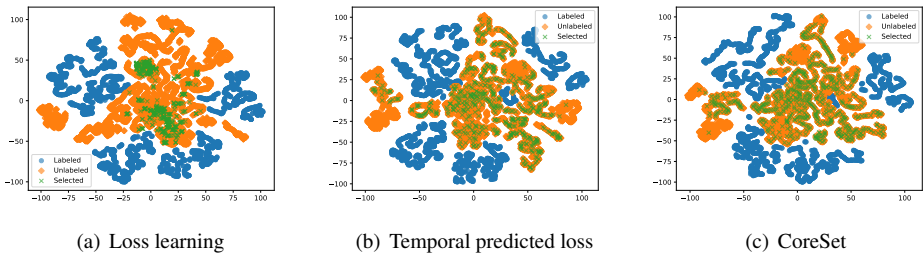(a) Loss learning   (b) Temporal predicted loss   (c) CoreSet

Figure 4: Diversity comparison using t-SNE with a perplexity of 30 for GTAVs route 2. TPL drastically increases the diversity compared to the vanilla approach, such that the diversity is comparable to CoreSet.

will omit $\mathcal{D}_i$ in further equations. As time is strictly increasing, the time derivative of the predicted loss can be defined by the derivative of the loss prediction module f_σ in Equation 2.

$$f_\sigma(x(t)|\mathcal{D}_i) = \tilde{\sigma}_x \approx \sigma_p. \quad (1) \qquad\qquad \frac{d\tilde{\sigma}_x}{dt} = \frac{d}{dt} f_\sigma(x(t)) = \frac{df_\sigma}{dx} \cdot \frac{dx}{dt} \quad (2)$$

Due to the discrete time steps in data streams, we calculate the derivative of $\frac{d}{dt}\tilde{\sigma}_x$ numerically. We estimate the temporal change of uncertainty via the derivative of the predicted loss $\tilde{\sigma}_x$ and use the temporal predicted loss (TPL) $\frac{d}{dt}\tilde{\sigma}_x$ to quantify the usefulness of a sample x.

Additionally, we modified the loss function of the loss prediction module. The ranking loss function proposed by Yoo and Kweon [37] is only capable of learning a relative loss order between two losses l_i and l_j of the task model and the predicted losses $\hat{l}_i$ and $\hat{l}_j$. Therefore, we added an L1-loss with an activation threshold and a scaling factor λ to smooth the learning of the loss and reduce the possible offsets between the task loss and its loss prediction. With the L1 part, we bound the linear offset of the predicted loss, which is impossible when only considering the ranking loss. The threshold avoids a noisy loss of the module based on the continuously changing loss of the task model.

$$\mathcal{L}_{\text{Loss}} = \frac{2}{B} \sum_{\substack{i=\text{odd} \\ j=\text{even}}}^{B} \max(0, \text{Sign}^*(l_i - l_j) \cdot (\hat{l}_i - \hat{l}_j) + \xi) + \frac{\lambda}{B} \cdot \sum_i^B \text{L1}_{\text{margin}}(l_i, \hat{l}_i, \zeta) \quad (3)$$

With $\text{L1}_{\text{margin}}$ being an L1 loss set to zero if the distance is below a given threshold ζ and B as batch size. The Sign^* function is modified from the standard sign function, where 0 is included as a negative number and not a positive number, as introduced by [37]. For the final loss, we combined the loss module loss scaled by a factor η with a cross-entropy loss, as proposed by [37].

5   Evaluation of Temporal Predicted Loss

To evaluate our approach, we compare TPL against the vanilla loss learning approach to show the increased diversity in Section 5.1, state-of-the-art stream-based approaches in Section 5.2 and state-of-the-art pool-based approaches in Section 5.3.

In each AL cycle, a recorded drive is either added to the pool or shown the model as a stream for the pool-stream or stream-batch scenario, respectively. Afterward, the model is

trained from scratch and evaluated before a new cycle begins. We did not split recordings to avoid overlaps in the train, validation and test split. Over the cycles, the model can select an additional 50% of the initial pool size as samples to the labeled pool. To reach this size, we selected 10% of the current stream for the GTAVs dataset and 20% for A2D2s. As the length of the recorded streams varies, a different number of samples is selected at each cycle.

The methods are evaluated by their accuracy against the percentage of the labeled data concerning the initial training set and unlabeled pool size. We employ a ResNet18 [17] model and a VGG11 [55] model with batch normalization, common in AL benchmarks. To make a fair comparison with sampling methods like BatchBald easier, we extended the classification head of the ResNet18 model to include three fully connected layers with dropout layers in between. The loss learning modules are attached after each of the four ResNet blocks and four max pooling layers for VGG11. As baselines, we trained the tested models using all the available training data (100%), a random selection strategy and methods introduced in the related work. We conducted a small hyperparameter search to select the appropriate settings to ensure comparability among the different methods. The supplementary Appendix C gives detailed information supporting reproducibility.

5.1 Temporal Coherence Leads to Batch Diversification

In our first experiment, we aim to investigate the impact of temporal relations on the selection of data samples using t-SNE analysis. We compare the results of loss learning [57], TPL without the loss function adaption in Equation 3 and the CoreSet [52] approach in a t-SNE analysis using the GTAVs dataset in Figure 4. Since CoreSet is a diversity-based method unsuited for streaming data, we use the pool-stream scenario but the stream-batch for loss learning and TPL. As shown in Figure 4(a), the results indicate that loss learning selects a specific cluster of uncertainty with low diversity in the latent space. In contrast, the TPL method, as shown in Figure 4(b), demonstrates a high level of diversity, which is slightly less diverse than the diversity-based CoreSet approach shown in Figure 4(c).

TPL successfully achieved its objective of enhancing batch diversity without requiring expensive distance calculations. Moreover, as it is a pure uncertainty-based method, it is well-suited for streaming data.
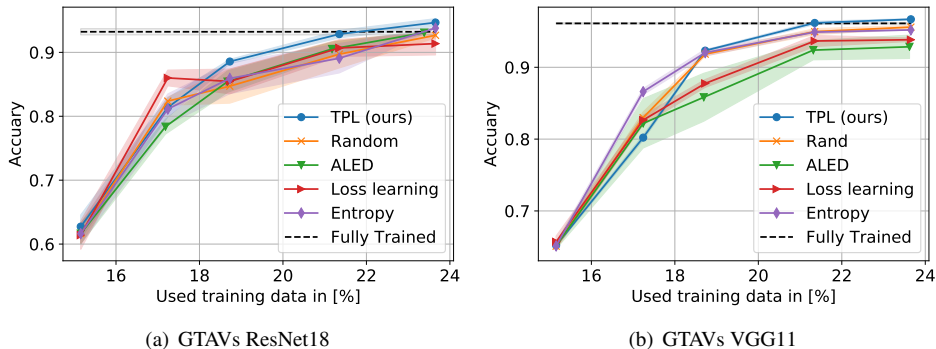

(a) GTAVs ResNet18

(b) GTAVs VGG11

Figure 5: Stream-based comparison - accuracy over used data with indicated standard errors.

5.2   Stream-Based Scenario Evaluation

In these experiments, we compare our TPL method introduced in Section 4 with state-of-the-art methods for batch stream-based AL using the stream-batch scenario from Figure 3(b). We compare TPL for the ResNet18 and VGG11 models on the A2D2s and the GTAVs datasets. As an extension, we used a Deeplabv3 [6] with ResNet34 backbone to evaluate our method on semantic segmentation labels of A2D2.

In Figure 5 and Figure 7(a), we compare our TPL with random, MC dropout entropy [14], loss learning [57] and ALED [53] for A2D2s and GTAVs for ResNet18 and VGG11 with batch normalization.

In Figure 5(a), we can see that our TPL method intersects the line of the fully trained network at around 21.3% of the total training data, while the other methods intersect the fully trained network's line between 23% and 23.8%. So for ResNet18, TPL achieves the same performance as the network train on all data with only 21% of the labels, saving up to 2.5pp more than other AL methods. For VGG11 in Figure 5(b), TPL intersects the fully trained networks again at 21% and converges slightly above the performance of the fully trained network. Interestingly, random is the second-best method for sample selection. A possible explanation is the data's high redundancy and temporal linearity. The A2D2s dataset, being small and based on real-world data, can result in more significant variations when selecting samples from other recordings. Additionally, the conditions of the stream used for selection may differ from those of the test and validation set (e.g., weather), which can lead to a decrease in performance with new data, as observed in Figure 7(a). It is possible to exceed the performance of fully-trained network, as the selected subsets may better represent the distribution of the test set than the full dataset. In Figure 7(a), TPL achieves the fully trained network's performance at 35.5% percent of all data. While ALED and random manage a head start before loss learning, the difference gets smaller at 36% of the data. With ALED intersecting at 37.5% and loss learning at 38.5%, TPL saves 2 pp more data than state-of-the-art methods.

To underline the flexibility of TPL, we also verify it on the semantic segmentation dataset part with the DeeplabV3 [6] model. The splits, query size and selection parameters are identical to the classification experiments. All details, including loss module attachment points, are given in Appendix D.

Figure 6 shows the limitation of stream-based AL created by the data logistic advantage. While the pool-based selection process can be continued until a desired performance is reached, the number of cycles for stream-based scenario is tied to the number of streams. Here, all methods fail to reach the fully trained network's performance. Selecting 20% of each stream is too little for semantic segmentation.

Figure 6:   Semantic segmentation comparison with indicated standard errors.

Notably, TPL achieves the highest mean Intersection over Union (mIoU) by using only 39% of the data. The mIoU value is 1% higher than the second-best methods without any parameter or adaptation. This proves the potential and flexibility to apply TPL to other tasks.
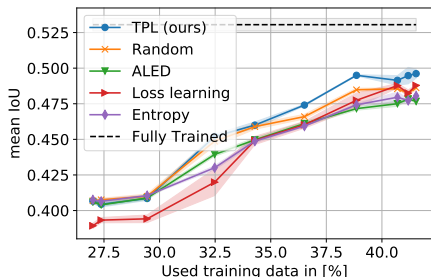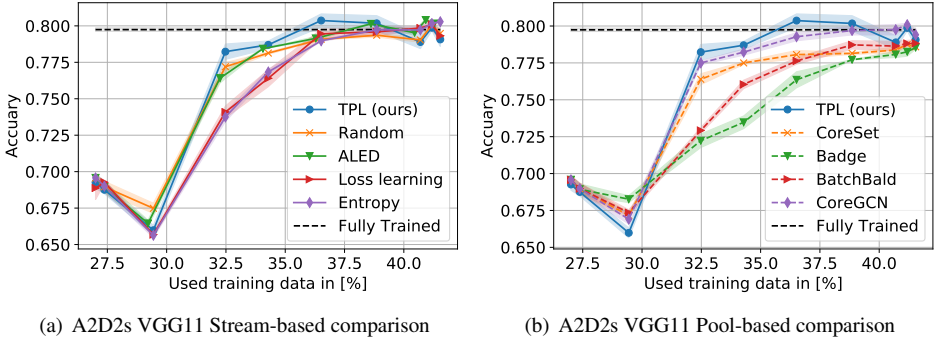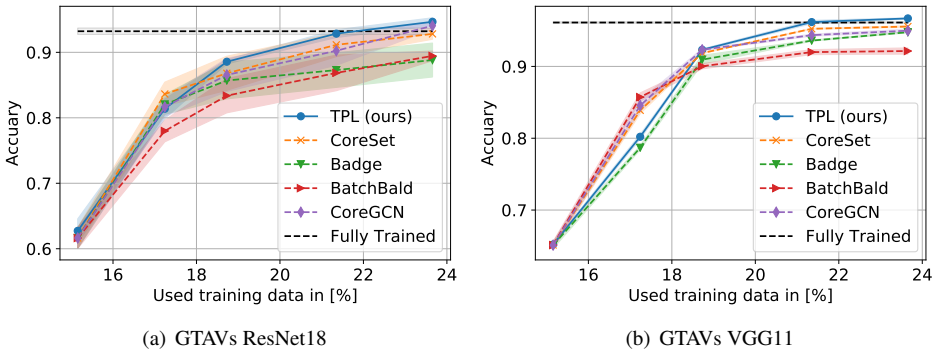
(a) A2D2s VGG11 Stream-based comparison

(b) A2D2s VGG11 Pool-based comparison

Figure 7: Stream-based and Pool-based comparison for A2D2 - accuracy over used data with indicated standard errors.

5.3 Comparing TPL with Pool-Based Scenarios

To introduce stream-based strategies as a genuine alternative to pool-based AL, we compare our TPL method with pool-based AL approaches. We use the stream-batch scenario for TPL and the pool-stream scenario for BatchBald [23], CoreSet [42], Badge [1] and CoreGCN [5]. To the best of our knowledge, we are the first to compare the different scenarios explicitly.

TPL outperforms the second-best method CoreGCN by reaching the fully trained network line with 3 pp fewer data for A2D2 with VGG and 2 pp fewer data for GTAVs with Resnet18 depicted in Figure 7(b) and Figure 8(a). However, the pool-based methods do not suffer from the limited selection of the last streams of the A2D2s. For GTAVs with VGG11 illustrated in Figure 8(b), TPL is the only approach crossing the fully trained line, while CoreGCN stays slightly behind. CoreSet and the learning-based CoreGCN considering diversity outperform the other pool-based methods. Although the pool-based scenario provides an information advantage, pool-based methods without diversity consideration show poor performance. Our results suggest that diversity is crucial in sample selection on highly redundant datasets.

(a) GTAVs ResNet18

(b) GTAVs VGG11

Figure 8: Pool-based comparison - accuracy over used data with indicated standard errors.

Additionally, we compared the selection times of the different methods on a dedicated machine. Table 1 shows our computational efficiency, especially against pool-based methods with at least four times. TPL offers data savings like state-of-the-art pool-based methods

while offering the reduced data logistics of stream-based approaches with an at least seven times faster selection. As most perception problems start from sensor streams (e.g., camera), our TPL method can be used on the mobile device directly for many applications like robotics, autonomous driving and environmental surveillance, saving additional computational costs and data-prepossessing efforts.

Method	Loss learn.	TPL	Entropy	ALED	BatchBald	Badge	CoreSet	CoreGCN
Time [s]	**4.5**	<u>4.6</u>	6.3	427.2	835.2	49.7	32.7	49.7

Table 1: Mean selection time on GTAVs/ResNet18 - Nvidia A6000, Intel Xeon Gold 6248R.

6   Conclusion and Future Work

Our work explored temporal coherence in data for stream-based active learning (AL) in perception. We proposed a novel approach leveraging temporal coherence to increase the diversity of uncertainty-based AL methods. We combined it with a modification of the loss learning module [57] in our temporal predictive loss (TPL) method. To reflect the properties of stream-based data collection, we introduced the A2D2 streets (A2D2s) and GTA V street (GTAVs) datasets and made them publicly available. Additionally, we introduced the pool-stream scenario to enable comparisons with the stream-batch scenario.

Our experiments showed that TPL significantly increases the diversity of the selection compared to vanilla loss learning and achieves almost the diversity of diversity-based methods. Compared to state-of-the-art stream-based AL methods, TPL achieves the highest accuracy with 2.5 pp less annotated data, using only 21% of the GTAVs dataset. To the best of our knowledge, our work provided the first comparison between stream-based and pool-based AL using pool-stream and stream-batch scenarios, showing that our TPL method outperforms pool-based methods while being seven times faster. Overall, this study represents an essential step towards stream-based AL for perception tasks, demonstrating the effectiveness of stream-based approaches. Furthermore, it highlights stream-based approaches as a viable alternative to pool-based approaches. The additional advantage in data logistics of stream-based AL is a significant consideration for enabling large-scale AL.

In future work, we plan to investigate alternative methods for uncertainty estimation. One potential avenue of research is integrating uncertainty estimation with diversity measurements, which have shown promising results in pool-based approaches. Additionally, we plan to extend our approach to exploit temporally coherent vision data in lane and object detection. Finally, a dedicated dataset for stream-based AL in mobile robot perception would further the field and provide valuable resources to the research community.

Acknowledgement

References

[1] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *Proceeding of International Conference on Learning Representations (ICLR)*, 2020.

[2] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Javad Zolfaghari Bengar, Abel Gonzalez-Garcia, Gabriel Villalonga, Bogdan Raducanu, Hamed H. Aghdam, Mikhail Mozerov, Antonio M. Lopez, and Joost van de Weijer. Temporal coherence for active learning in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[4] Davide Cacciarelli, Murat Kulahci, and John Sølve Tyssedal. Stream-based active learning with linear models. *Knowledge-Based Systems*, 254, 10 2022. ISSN 09507051. doi: 10.1016/j.knosys.2022.109664.

[5] 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.

[6] 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.

[7] Ioannis Chiotellis, Franziska Zimmermann, Daniel Cremers, and Rudolph Triebel. Incremental semi-supervised learning from streams for object classification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5743–5749, 2018.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, Daimler Ag R&d, and T U Darmstadt. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL www.cityscapes-dataset.net.

[9] 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.

[10] 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.

[11] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. Deep active learning for efficient training of a lidar 3d object detector. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 667–674, 2019.

[12] Jianxiang Feng, Jongseok Lee, Maximilian Durner, and Rudolph Triebel. Bayesian active learning for sim-to-real robotic perception. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[13] Kaito Fujii and Hisashi Kashima. Budgeted stream-based active learning via adaptive submodular maximization. In *Proceedings of Neural Information Processing Systems*, 2016.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *In Proceedings of the 33th International Conference on International Conference on Machine Learning (ICML)*, page 1050–1059, 2016.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. URL www.cvlibs.net/datasets/kitti.

[16] 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.

[17] 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.

[18] Aral Hekimoglu, Michael Schmidt, Alvaro Marcos-Ramiro, and Gerhard Rigoll. Efficient active learning strategies for monocular 3d object detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[19] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv*, 1112.5745, 2011.

[20] Po Yu Huang, Wan Ting Hsu, Chun Yueh Chiu, Ting Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[21] Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3311–3320. PMLR, 2019.

[22] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8166–8175, 2021.

[23] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7026–7037, 2019.

[24] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[25] Alexander Narr, Rudolph Triebel, and Daniel Cremers. Stream-based active learning for efficient and adaptive classification of 3d objects. In *proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[26] 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.

[27] Tuan Pham, Daniel Kottke, Georg Krempl, and Bernhard Sick. Stream-based active learning for sliding windows under the influence of verification latency. *Machine Learning*, 111(6):2011–2036, Jun 2022. ISSN 1573-0565. doi: 10.1007/s10994-021-06099-z. URL https://doi.org/10.1007/s10994-021-06099-z.

[28] L. Puck, N. Spielbauer, D. Schable, T. Schnell, T. Buttner, A. Roennau, and R. Dillmann. Distributed active learning for semantic segmentation on walking robots. In *Proceedings of the 20th International Conference on Advanced Robotics (ICAR)*, pages 461–467. Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781665436847. doi: 10.1109/ICAR53236.2021.9659389.

[29] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54:1 – 40, 2022.

[30] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016-December, pages 3234–3243, 12 2016.

[31] 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.

[32] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 8 2018.

[33] Yuya Senzaki and Christian Hamelain. Active learning for deep neural networks on edge devices. *ArXiv*, 2106.10836, 2021.

[34] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2010.

[35] 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.

[36] 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.

[37] 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.

[38] Łukasz Korycki, Alberto Cano, and Bartosz Krawczyk. Active learning with abstaining classifiers for imbalanced drifting data streams. In *Proceedings of the IEEE International Conference on Big Data*, 2019.