Label Efficient Visual Abstractions for Autonomous Driving

Aseem Behl*[1,2], Kashyap Chitta*[1,2], Aditya Prakash[1], Eshed Ohn-Bar[1,3] and Andreas Geiger[1,2]

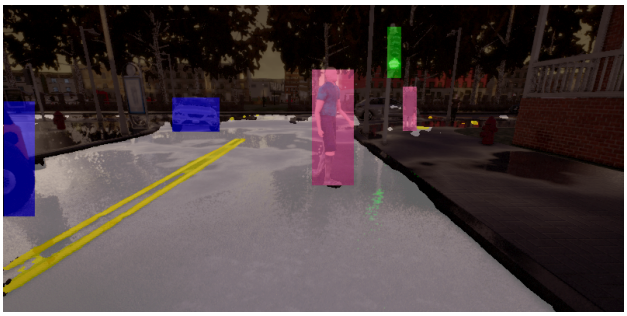
***Abstract*— It is well known that semantic segmentation can be used as an effective intermediate representation for learning driving policies. However, the task of street scene semantic segmentation requires expensive annotations. Furthermore, segmentation algorithms are often trained irrespective of the actual driving task, using auxiliary image-space loss functions which are not guaranteed to maximize driving metrics such as safety or distance traveled per intervention. In this work, we seek to quantify the impact of reducing segmentation annotation costs on learned behavior cloning agents. We analyze several segmentation-based intermediate representations. We use these *visual abstractions* to systematically study the trade-off between annotation efficiency and driving performance, i.e., the types of classes labeled, the number of image samples used to learn the visual abstraction model, and their granularity (e.g., object masks vs. 2D bounding boxes). Our analysis uncovers several practical insights into how segmentation-based visual abstractions can be exploited in a more label efficient manner. Surprisingly, we find that state-of-the-art driving performance can be achieved with orders of magnitude reduction in annotation cost. Beyond label efficiency, we find several additional training benefits when leveraging visual abstractions, such as a significant reduction in the variance of the learned policy when compared to state-of-the-art end-to-end driving models.**

I. Introduction

Significant research effort has been devoted into semantic segmentation of street scenes in recent years, where images from a camera sensor mounted on a vehicle are segmented into classes such as road, sidewalk, and pedestrian [2], [7], [10], [13], [32]. It is widely believed that this kind of accurate scene understanding is key for robust self-driving vehicles. Existing state-of-the-art methods [35] optimize for image-level metrics such as mIoU, which is challenging as it requires a combination of coarse contextual reasoning and fine pixel-level accuracy [9]. The emphasis on such image-level requirements has resulted in large segmentation benchmarks, i.e., thousands of images, with high labeling costs. However, the development of such benchmarks, in terms of annotation type and cost, is often done independently of the actual *driving task* which encompasses optimizing metrics such as distance traveled per human intervention.

In parallel, there has been a surge in interest on using *visual priors* for learning end-to-end control policies with improved performance, generalization and sample efficiency [17], [26], [36]. Instead of learning to act directly from image observations, which is challenging due to the high-dimensional input, a visual prior is enforced by decomposing the task into intermediate sub-tasks. These intermediate visual sub-tasks, e.g., object detection, segmentation, depth and motion estimation, optimized independently, are then fed as an input to a policy learning algorithm, e.g., [26]. In particular, semantic segmentation has been shown to act as a powerful visual prior for driving agents [18], [27]. While beneficial for learning robust policies, such intermediate sub-tasks require explicit supervision in the form of additional manual annotations. For several visual priors, obtaining these annotations can be time consuming, tedious, and prohibitive.

Trained with 6400 finely annotated images and 14 classes
Annotation time ≈ 7500 hours, policy success rate = 50%



Trained with 1600 coarsely annotated images and 6 classes
Annotation time ≈ 50 hours, policy success rate = 58%

Fig. 1. **Label efficient visual abstractions for learning driving policies.** To address issues with obtaining time-consuming annotations, we analyze image-based representations that are both ***efficient*** in terms of annotation cost (e.g., bounding boxes), and ***effective*** when used as intermediate representations for learning a robust driving policy. Considering six coarse safety-critical semantic categories and combining non-salient classes (e.g., sidewalk and building) into a single class can significantly reduce annotation cost while at the same time resulting in more robust driving performance.

A useful visual prior needs to encode the right assumptions about the environment in order to simplify policy learning. In the case of autonomous driving, semantic segmentation encodes the fact that certain pixels in an image can be treated similarly: e.g. the agent can drive on roads but not sidewalks; the agent must not collide with other vehicles or pedestrians. However, it is unclear which semantic classes are relevant to the driving task and to which granularity they should be labeled. This motivates our study of *visual abstractions*,

* indicates equal contribution, listed in alphabetical order. [1]Max Planck Institute for Intelligent Systems, Tübingen; [2]University of Tübingen; [3]Boston University. {firstname.lastname}@tue.mpg.de

which are compact semantic segmentation-based representations of the scene with fewer classes, coarser annotation, and learned with little supervision (only several hundred images). We consider the following question: when used as visual priors for policy learning, are representations obtained from datasets with lower annotation costs competitive in terms of driving ability?

Towards addressing this research question, we systematically analyze the performance of varying intermediate representations on the recent NoCrash benchmark of the CARLA urban driving simulator [6], [8]. Our analysis uncovers several new results regarding label efficient representations. Surprisingly, we find that certain visual abstractions learned with only a fraction of the original labeling cost can still perform as well or better when used as inputs for training behavior cloning policies (see Fig. 1). Overall, our contributions are three-fold:

- Given the same amount of training data, we empirically show that using classes less relevant to the driving policy can lead to degraded performance. We find that only few of the commonly used classes are directly relevant for the driving task.
- We demonstrate that despite requiring only a few hundred annotated images in addition to the expert driving demonstrations, training a behavior cloning policy with visual abstractions can significantly outperform methods which learn to drive from raw images, as well as existing state-of-the-art methods that require a prohibitive amount of supervision.
- We further show that our visual abstractions lead to a large variance reduction when varying the training seed which has been identified as a challenging problem in imitation learning [6].

Our code is available at `https://github.com/autonomousvision/visual_abstractions`

II. Related Work

This work relates to visual priors for robotic control and behavior cloning methods for autonomous driving. In this section, we briefly review the most related works.

Semantic Segmentation: Segmentation of street scenes has received increased interest in robotics and computer vision due to its implications for autonomous vehicles, with several benchmarks and approaches released in recent years [7], [13], [32]. Progress has been achieved primarily through methods that use supervised learning, with architectural innovations that improve both contextual reasoning and fine pixel-level details [16], [31], [35]. However, generating high-quality ground truth to build semantic segmentation benchmarks is a time-consuming and expensive task. For instance, labeling a single image was reported to take 90 minutes on average for the Cityscapes dataset [7], and approximately 60 minutes for the CamVid dataset [2]. Our work focuses on reducing demands for annotation quality and quantity, which is important in the context of reducing annotation costs for segmentation and autonomous driving.

Behavior Cloning for Autonomous Driving: Behavior cloning approaches learn to map sensor observations to desired driving behavior through supervised learning. Behavior cloning for driving has historical roots [20] as well as recent successes [1], [19], [21], [33]. Bojarski et al. [1] propose an end-to-end CNN for lane following that maps images from the front facing camera of a car to steering angles, given expert data. Conditional Imitation Learning (CIL) extends this framework by incorporating high-level navigational commands into the decision making process [5]. Codevilla et al. [6] present an analysis of several limitations of CIL. In particular, they observe that driving performance drops significantly in new environments and weather conditions. They also observe drastic variance in performance caused by model initialization and data sampling during training. The goal of this work is to address these issues with semantic input representations while maintaining low labeling costs.

Visual Priors for Improving Generalization: Recent papers have shown the effectiveness of using mid-level visual priors to improve the generalization of visuomotor policies [17], [18], [26], [29], [36]. Object detection, semantic segmentation and instance segmentation have been shown to help significantly for generalization in navigation tasks [29], [36]. Müller et al. [18] train a policy in the CARLA simulator with a binary road segmentation as the perception input, demonstrating that learning a policy independent of the perception and low-level control eases the transfer of learned lane-keeping behavior for empty roads from simulation to a real toy car. More recently, Zhao et al. [34] and Toromanoff et al. [28] show how to effectively incorporate knowledge from segmentation labels into a behavior cloning and reinforcement learning network, respectively. These existing studies either compare different visual priors or focus on improving policies by choosing a specific visual prior, regardless of the annotation costs. Nonetheless, knowing these costs is extremely valuable from a practitioner's perspective. In this work, we are interested in identifying and evaluating label efficient representations in terms of the performance and variance of the learned policies.

Compact Representations for Driving: Instead of using a comparably higher-dimensional visual prior such as pixel-level segmentation, Chen et al. [3] present an approach which estimates a small number of human interpretable, predefined affordance measures such as the angle of the car relative to the road and the distance to other cars. These predicted affordances are then mapped to actions using a rule-based controller, enabling autonomous driving in the TORCS racing car simulator [30]. Similarly, Sauer et al. [25] estimate several affordances from sensor inputs to drive in the CARLA simulator. In contrast to [3], they consider the more challenging scenario of urban driving where the agent needs to avoid collision with obstacles on the road and navigate junctions with multiple possible driving directions. They achieve this by expanding the set of affordances to be more applicable to urban driving. These methods are relevant to our study, in that they simplify perception through

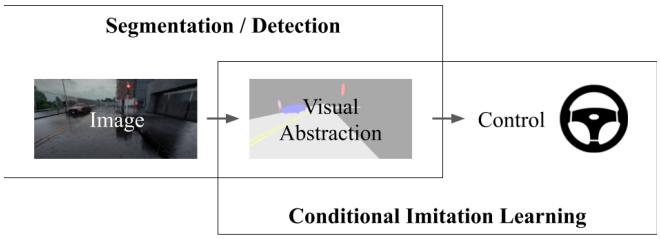


Fig. 2. **High-level overview of the proposed study.** We investigate different segmentation-based visual abstractions by pairing them with a conditional imitation learning framework for autonomous driving.

compact representations. However, these affordances are hand-engineered and very low-dimensional. Thus, failures in design will lead to errors that cannot be recovered from.

III. Method

As illustrated in Fig. 2, we consider a modular approach that comprises two learned mappings, one from the RGB image to a semantic label map and one from the semantic label map to control. To learn these mappings, we use two image-based datasets, (i) $S = \{\mathbf{x}^i, \mathbf{s}^i\}_{i=1}^{n_s}$ which consists of n_s images annotated with semantic labels, and (ii) $C = \{\mathbf{x}^i, \mathbf{c}^i\}_{i=1}^{n_c}$ which consists of n_c images annotated with expert driving controls. First, we train the parameters of a visual abstraction model a_ϕ parameterized by ϕ using the segmentation dataset S. The trained visual abstraction stack is then applied to transform C resulting in a control dataset $C_\phi = \{a_\phi(\mathbf{x}^i), \mathbf{c}^i\}_{i=1}^{n_c}$ on which we train a driving policy π_θ with parameters θ. At test time, control values are obtained for an image $\mathbf{x}^*$ by composing the two learned mappings, $\mathbf{c}^* = \pi_\theta(a_\phi(\mathbf{x}^*))$.

In this section, we discuss the core questions we aim to answer, followed by a description of the visual abstractions and driving agent considered in our study.

A. Research Questions

We aim to build a segmentation dataset S that is cost-effective, yet encodes all relevant information for policy learning. We are interested in the following questions:

Can selecting specific classes ease policy learning? A semantic segmentation $\mathbf{s}$ assigns each pixel to a discrete category $k \in \{1, \ldots, K\}$. Knowing whether a pixel belongs to the building class or tree class may provide no additional information to a driving agent, if it knows that the pixel does not belong to the road, vehicle or pedestrian class. We are interested in understanding the impact of the set of categories on the driving task.

Are semantic representations trained with few images competitive? In a policy learning setting, the training of the driving agent may be able to automatically compensate for some drop in performance of the segmentation model. We aim to determine if a parsimonious training dataset obtained by reducing the number of training images n_s for the segmentation model can achieve satisfactory performance.

Is fine-grained annotation important? Exploiting coarse annotations such as 2D bounding boxes instead of pixel-accurate segmentation masks can alleviate the key challenge in building segmentation models: annotation costs [37]. If fine-grained annotation can be avoided, we are interested in how to select a_ϕ to exploit coarse annotation during training.

Are visual abstractions able to reduce the variance which is typically observed when training agents using behavior cloning? Significant difference in performance of behavior cloning policies is caused as a result of changing the training seed or the sampling of the training data [6]. This is problematic in the context of autonomous driving where evaluating an agent is expensive and time-consuming, making it difficult to assess if changes in performance are a result of algorithmic improvements or random training seeds. Since visual priors abstract out certain aspects of the input such as illumination and weather, we are interested in investigating their effect on reducing the variance in policies with different random training seeds.

B. Visual Abstractions

For our analysis, we consider three visual abstractions based on semantic segmentation.

Privileged Segmentation: As an upper bound, the ground-truth semantic labels (available from the simulator) can be used directly as an input to the driving agent. This form of privileged information is useful for ablative analysis.

Standard Segmentation: For standard pixel-wise segmentation over all classes, our perception stack is based on a ResNet and Feature Pyramid Network (FPN) backbone [11], [14], with a fully-convolutional segmentation head [16].

Hybrid Detection and Segmentation: To exploit coarser annotation, we use a hybrid architecture that distinguishes between stuff and thing classes [12]. The architecture consists of a segmentation model trained on stuff classes annotated with semantic labels (e.g. road, lane marking) and a detection model based on Faster-RCNN [22] trained on thing classes annotated with bounding boxes (e.g. pedestrian, vehicle). The final visual abstraction per pixel is obtained by overlaying the pixels of detected bounding boxes on top of the predicted segmentation, based on a pre-defined class priority. Similar hybrid architectures have been found useful previously in the urban street scene semantic segmentation setting, since detectors typically have different inductive biases than segmentation networks [24].

C. Driving Agent

Conditional Imitation Learning: In general, behavior cloning involves a supervised learning method which is used to learn a mapping from observations to expert actions. However, sensor input alone is not always sufficient to infer optimal control. For example, at intersections, whether the car should turn or keep straight cannot be inferred from camera images alone without conditioning on the goal. We therefore follow [5], [6] and condition on a navigational command. The navigational command represents driver intentions such as the direction to follow at the next intersection. Our agent is a neural network π_θ with parameters θ, which maps a semantic

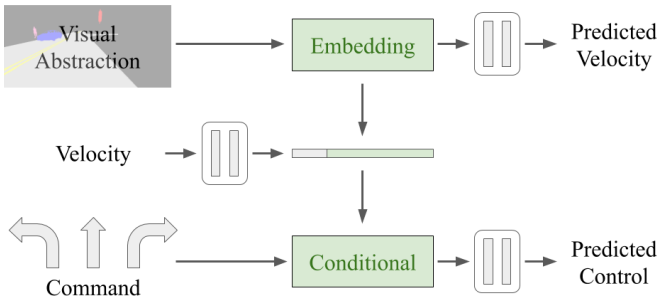


Fig. 3. **Driving agent architecture.** Given a segmentation-based visual abstraction, current vehicle velocity, and discrete navigational command, the CILRS model predicts a control value [6].

representation $\mathbf{s} \in \mathcal{S}$, navigational command $\mathbf{n} \in \mathcal{N}$, and measured velocity $v \in \mathbb{R}^+$ to a control value $\mathbf{c} \in \mathcal{C}$:

$$\pi_\theta : \mathcal{S} \times \mathcal{N} \times \mathbb{R}^+ \rightarrow \mathcal{C} \tag{1}$$

Imitation Loss: In order to learn the policy parameters θ, we minimize an imitation loss $\mathcal{L}_{\text{imitation}}$ defined as follows:

$$\mathcal{L}_{\text{imitation}} = ||\mathbf{c} - \hat{\mathbf{c}}||_1 \tag{2}$$

Here, $\hat{\mathbf{c}} = \pi_\theta(\mathbf{s}, \mathbf{n}, v)$ is the control value predicted by our agent and $||\cdot||_1$ denotes the L_1 norm.

Velocity Loss: Recordings of expert drivers have an inherent inertia bias, where most of the samples with low velocity also have low acceleration. It is critical to not overly correlate these since the vehicle would prefer to never start after slowing down. As demonstrated in [6], predicting the current vehicle velocity as auxiliary task can alleviate this issue. We thus also use a velocity prediction loss:

$$\mathcal{L}_{\text{velocity}} = ||v - \hat{v}||_1 \tag{3}$$

Architecture: The architecture used for our driving agent is based on the CILRS model [6], summarized in Fig. 3. The visual abstraction is initially processed by an embedding branch, which typically consists of several convolutional layers. We flatten the output of the embedding branch and combine it with the measured vehicle velocity v using fully-connected layers. Since the space of navigational commands is typically discrete for driving, we use a conditional module to select one of several command branches based on the input command. The command branch outputs control values. Additionally, the output of the embedding branch is used for predicting the current vehicle speed, which is compared to the actual vehicle speed in the velocity loss $\mathcal{L}_{\text{velocity}}$ defined in Eq. 3. The final loss function for training is a weighted sum of the two components, with a scalar weight λ:

$$\mathcal{L} = \mathcal{L}_{\text{imitiation}} + \lambda \mathcal{L}_{\text{velocity}} \tag{4}$$

IV. Experiments

In this section, we present a series of experiments on the open-source driving simulator CARLA [8] to answer the questions raised in Section III-A. We perform our analysis by training and evaluating driving agents on the NoCrash benchmark of CARLA (version 0.8.4) [6].

TABLE I

Summary of control datasets. The third column indicates the number of labeled images used for training our detection and segmentation models. The fourth column indicates the approximate cost of annotating these images.

Name	Classes	Labeled Images	Cost (Hours)
Standard-Large-14	14	6400	7500
Standard-Large	6	6400	3200
Standard	6	1600	800
Standard-Small	6	400	200
Hybrid	6	1600	50

A. Task

The CARLA simulation environment consists of two map layouts, called Town 01 & Town 02, which can be augmented by 14 weather conditions. We use the provided 'autopilot' expert mode for data collection. All training and validation data is collected in Town 01 with the four weather conditions specified as the 'train' conditions in the NoCrash benchmark. The number of external agents is uniformly sampled from the range $[80, 160]$. Town 02 is reserved for testing.

Evaluation: The primary evaluation criterion in CARLA is the percentage of successfully completed episodes during an evaluation phase, referred to as Success Rate (SR). Successful navigation requires driving the vehicle from a starting position to a specified destination within a fixed time limit. Failure can be a result of collision with pedestrians, vehicles or static objects; or inability to reach the destination within the time limit (timeout). The benchmark consists of three levels of traffic density: Empty, Regular and Dense involving 0, 65 and 220 external agents respectively. We perform evaluation in Town 02 for two 'test' weather conditions of the NoCrash benchmark that are unseen during training.

B. Datasets

Perception: We collect a training set of 6400 images from Town 01 for training our detection and segmentation models. The images are annotated with 2D semantic labels for 14 classes, and 2D object boxes for 4 of these classes. These annotations are provided by the CARLA simulator.

Control: Following [6], we collect approximately 10 hours of driving frames and corresponding expert controls for imitation learning using the autopilot of the CARLA simulator. The images are sampled from three cameras facing different directions (left, center and right) on the car at 10 frames per second. We create five variants of this control dataset (summarized in Table I) by transforming the input RGB images to visual abstractions. The Standard-Large-14 dataset is generated using a segmentation network trained to segment the input into fourteen semantic classes: 'road', 'lane marking', 'vehicle', 'pedestrian', 'green light', 'red light', 'sidewalk', 'building', 'fence', 'pole','vegetation', 'wall', 'traffic sign' and 'other'. The remaining datasets, namely Standard-Large, Standard, Standard-Small, and Hybrid, use a reduced set of six classes: 'road', 'lane marking', 'vehicle', 'pedestrian', 'green light' and 'red light'. While the Standard datasets are generated using a segmentation network, the

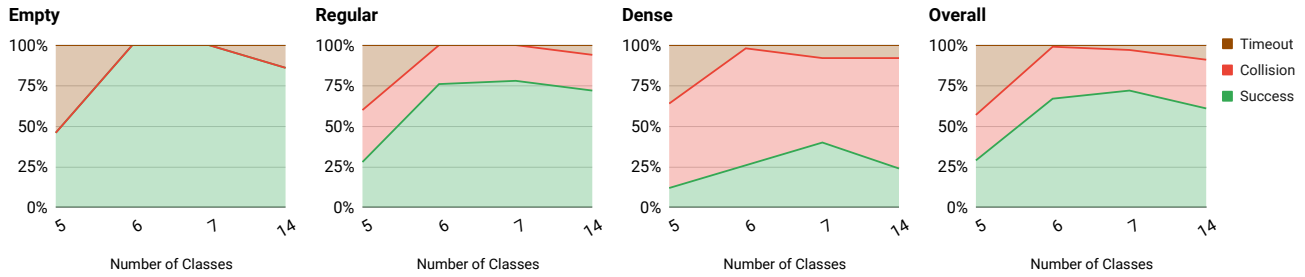


Fig. 4. **Identifying most relevant classes.** Success/collision/timeout percentages on the test environment (Town 02 Test Weather) of the CARLA NoCrash benchmark. For this ablation study, we use ground truth segmentation as inputs to the behavior cloning agent. Reduction from fourteen to seven or six classes leads to a slight increase in success rate, but further reduction to five classes leads to a large number of failures.

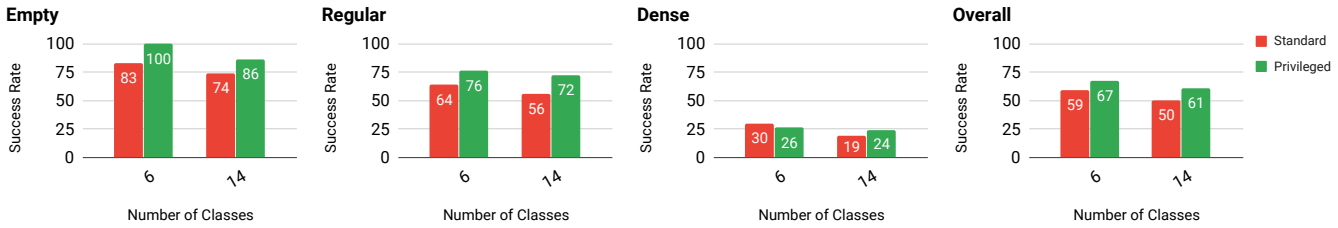


Fig. 5. **Evaluating the six-class representation.** Success Rate on the test environment (Town 02 Test Weather) of the CARLA NoCrash benchmark. The six-class representation consistently leads to better performance than the fourteen-class representation while simultaneously having lower annotation costs.

Hybrid dataset is generated using the combination of a 2-class segmentation model and 4-class detection model.

The study of [37] provides approximate labeling costs based on a labeling error threshold, for a dataset of similar visual detail as ours. The annotation time reported for the Standard abstraction (fine labeling) is around 300 seconds per image and per class. Our Hybrid visual abstraction roughly corresponds to a 32-pixel labeling error, which requires approximately 20 seconds per image and per class. Based on these statistics, we include the estimated annotation time for each visual abstraction in Table I.

In addition, we also collect a Privileged dataset comprising the ground truth semantic segmentation for the input, which we use for ablation studies involving privileged agents.

C. Implementation Details

Our perception models are based on a ResNet-50 FPN backbone pre-trained on the MS-COCO dataset [15]. We finetune this model for 3k iterations on the perception dataset with the hyper-parameters set to the default of the Detectron2[1] detection and segmentation algorithms.

The driving agents use a ResNet-18 model in the 'embedding' branch (see Fig. 2). We process the velocity input with two fully-connected layers of 128 units each, which is combined with the ResNet output in another fully-connected layer of 512 units. The velocity prediction branch and each command branch encompass two fully-connected layers of 256 units. We train each model from scratch for 200k iterations using the default training hyper-parameters of the COiLTRAiNE[2] framework. This is the standardized repository used to train imitation learning agents on CARLA [6], [34], which currently supports CARLA version 0.8.4.

[1]https://github.com/facebookresearch/detectron2
[2]https://github.com/felipecode/coiltraine

D. Results

Identifying Most Relevant Classes: In our first experiment, our goal is to analyze the impact of training policies with a reduced set of annotated classes. For this, we train and evaluate agents using the Privileged dataset. As a baseline, we use a semantic representation consisting of all fourteen classes in the dataset. We then evaluate a reduced subset of seven classes that we hypothesize to be the most relevant: 'road', 'sidewalk', 'lane marking', 'vehicle', 'pedestrian', 'green light' and 'red light'. Next, we evaluate representations that consist of six and five classes, by excluding 'sidewalk' and then 'lane marking' from the seven-class representation. For the five-class representation, we re-label 'lane marking' as 'road'. The driving performance for these representations is summarized in Fig. 4. We show the percentage of evaluation episodes in the test environment where the agent succeeded, collided with an obstacle or timed out.

We note from our results that perfect segmentation accuracy does not mean perfect overall perception. The fourteen-class model does not achieve perfect driving in any of the three traffic conditions. Even with perfect perception, limitations of using behavior cloning methods such as covariate shift, where the states encountered at train time differ from those encountered at test time, can lead to non-optimal driving behavior. Further, the higher relative dimensionality when using fourteen classes, which includes fine details of classes such as fences, buildings, and vegetation, makes it harder for the agent to identify the right features important for generalization. This is reflected by the fact that the seven-class representation outperforms the agent based on

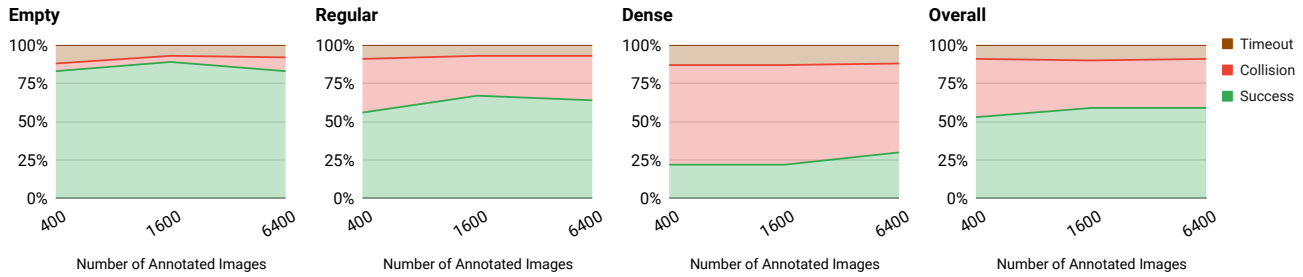


Fig. 6. **Comparing visual abstractions as annotation quantity is reduced.** Success/collision/timeout percentages on the test environment (Town 02 Test Weather). Mean over 5 random training seeds. Performance remains consistent with 6400 or 1600 annotated images, with a slight drop as the training dataset for the visual abstraction is reduced to 400 images.

fourteen classes in all three traffic conditions. We empirically observe that the fourteen-class agent is more conservative in its driving style, and more susceptible to timeouts.

The six-class representation that excludes sidewalk segmentation achieves similar performance to seven classes in empty and regular traffic. We therefore additionally compare the six-class and fourteen-class representations using inferred visual abstractions without privileged information, in order to analyze if the same trends observed in Fig. 4 hold. Specifically, we compare the Standard-Large-14 and Standard-Large datasets as described in Table I. These datasets are generated using fourteen-class and six-class segmentation networks respectively. The success rates of these trained models are shown in Fig. 5. Additionally, we show the performance of the corresponding six-class and fourteen-class Privileged agents for reference. We observe that the six-class representation consistently maintains or improves upon the performance of agents trained using all fourteen classes. The six-class approach helps to reduce annotation costs by removing the requirement of assigning labels to several classes such as poles and vegetation, which can be time-consuming due to thin structures with a lot of fine detail.

Interestingly, we observe from Fig. 4 that using only five classes leads to a significant reduction in performance, with the overall success rate dropping from 67% to 29%. This drastic change indicates that the lane marking class is of very high importance for learning driving policies, and the task becomes hard to solve without this class even with perfect segmentation accuracy on all other classes. Based on the consistent performance of the six-class visual abstraction in both Fig. 4 and Fig. 5, we choose this representation to perform a more detailed analysis of trade-offs related to labeling quality and quantity.

Number of Annotated Images: In our second experiment, we study the impact of reducing annotation quantity by training agents using the Standard-Large, Standard, and Standard-Small datasets from Table I. Reducing from Standard-Large to Standard-Small, each dataset has 4 times less samples (and therefore 4 times less labeling cost) than the preceding one. Our results, presented as the mean success, collision and timeout percentages over 5 different training seeds for the behavior cloning agent, are summarized in Fig. 6.

We observe no significant differences in overall downstream driving task performance between the agents trained on 6400 or 1600 samples, and a slight drop when using 400 images. Taking a closer look at the driving performance, we observe that the number of collisions in dense traffic is slightly lower for 6400 samples, but success rate is also slightly decreased on empty conditions. This shows that for our task, when focusing on only the most salient classes, a few hundred images are sufficient to obtain robust driving policies. In contrast, prior work that exploits semantic information on CARLA uses fine-grained annotation for several hours of driving data (up to millions of images) [34].

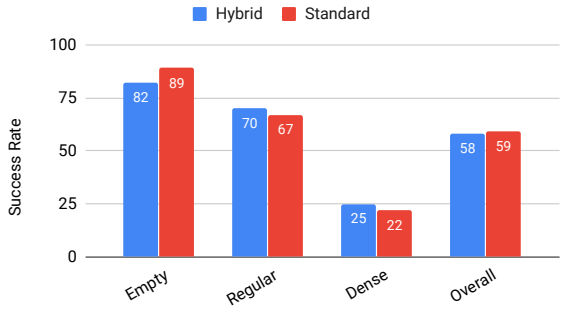


Fig. 7. **Evaluating the Hybrid visual abstraction.** Success rate on the test environment (Town 02 Test Weather) as the quality of annotation is reduced. Mean over 5 random training seeds. Overall, the performance of the Hybrid abstraction matches Standard segmentation despite having a reduction in annotation costs of several orders of magnitude.

From Fig. 6, we clearly observe a saturation in overall performance beyond 1600 training samples. We therefore study the impact of granularity of annotation in more detail while fixing the dataset size to 1600 samples.

Coarse Annotation: In our third experiment, we analyze the impact of using the Hybrid visual abstraction that utilizes coarse supervision during training, requiring only an estimated 50 hours to annotate. We present a comparison of the Hybrid and Standard visual abstractions in Fig. 7. The results are presented as the mean driving success rate of five training seeds. We find the Hybrid abstraction to improve performance on tasks involving external dynamic agents, i.e., regular and dense settings. Since objects are input as rectangles, without additional noise introduced by a standard segmentation network, we hypothesize that Hybrid abstractions are able to simplify policy learning (see Fig. 1). Overall, the performance of the Hybrid agent is on par with

TABLE II

VARIANCE BETWEEN RANDOM TRAINING SEEDS. PERCENTAGE SUCCESS RATE ON TOWN 02 TEST WEATHER FOR FIVE TRAINING SEEDS ON EMPTY (E), REGULAR (R), AND DENSE (D) CONDITIONS, AS WELL AS THE AVERAGE OVERALL (O) SUCCESS RATE. MAX AND MIN VALUES INDICATED IN **BOLD**. OUR APPROACH SIGNIFICANTLY REDUCES THE STANDARD DEVIATION AND COEFFICIENT OF VARIATION (CV).

Task	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Mean ↑	Std ↓	CV ↓
CILRS [6]								
E	**26**	44	42	**48**	46	41.20	8.79	0.21
R	**24**	26	30	32	**40**	30.40	6.23	0.20
D	**0**	2	4	4	**18**	5.60	7.13	1.27
O	**17**	24	25	28	**34**	25.60	6.18	0.24
Hybrid								
E	**76**	80	82	78	**90**	81.20	5.40	0.06
R	**64**	68	72	72	**72**	69.60	3.57	0.05
D	28	22	**18**	**34**	22	24.80	6.26	0.25
O	**55**	56	57	61	**61**	58.00	2.82	0.04

that of the Standard agent despite having approximately 15 times lower annotation costs (see Table I).

Variance Between Training Runs: In our fourth experiment, we investigate the impact of semantic representations on the variance between results for different training runs. In order to conduct a fair comparison, we use the same raw dataset for training all the models in this study. This ensures that there is no variance caused by the training data distribution. Similar to [6], the raw training data was collected by the standard CARLA data-collector framework[3].

We compare our approach to CILRS [6], which uses the weights of a network pre-trained on ImageNet [23] to reduce variance due to random initialization of the policy parameters. The only remaining source of variance in training is the random sampling of data mini-batches that occurs during stochastic gradient descent. However, existing studies have still reported high variance in CILRS models between training runs [6]. For our approach, we choose the agent trained with the Hybrid abstraction. The results of five different training seeds along with the mean, standard deviation and coefficient of variation (standard deviation normalized by the mean) for each method are shown in Table II.

We observe that for CILRS, the best training seed has double the average success rate of the worst training seed, leading to extremely large variance. In particular, on dense traffic conditions, the success rate ranges from 0 to 18. This amount of variance is problematic when trying to analyze or compare different approaches. In contrast, there is less variance observed when training with the Hybrid visual abstraction. Specifically, there is a significant reduction in the standard deviation across all traffic conditions, and the coefficient of variation is reduced by an order of magnitude.

We would like to emphasize that the variance reported in Table II comes from *training* with different random seeds. The training seed is the primary cause of variance, in addition to secondary evaluation variance which is caused by the random dynamics in the simulator. The existing practice

[3]https://github.com/carla-simulator/data-collector

TABLE III

COMPARISON TO STATE-OF-THE-ART. PERCENTAGE SUCCESS RATE ON TOWN 02 OF THE CARLA NOCRASH BENCHMARK, PRESENTED AS MEAN AND STANDARD DEVIATION OVER THREE EVALUATIONS OF THE SAME MODEL FOR EMPTY (E), REGULAR (R), AND DENSE (D) CONDITIONS. * INDICATES OUR RERUN OF THE BEST MODEL PROVIDED BY THE AUTHORS OF [6]. MODELS TRAINED WITH VISUAL ABSTRACTIONS OBTAIN STATE-OF-THE-ART RESULTS.

Task	CAL	CILRS*	LaTeS	LSD	Standard	Hybrid	Expert
Train Weather							
E	36 ± 3	65 ± 2	92 ± 1	$\mathbf{94 \pm 1}$	91 ± 2	87 ± 1	96 ± 0
R	26 ± 2	46 ± 2	74 ± 2	68 ± 2	77 ± 1	$\mathbf{82 \pm 1}$	91 ± 0
D	9 ± 1	20 ± 1	29 ± 3	30 ± 4	27 ± 7	$\mathbf{41 \pm 1}$	41 ± 2
Test Weather							
E	25 ± 3	71 ± 2	83 ± 1	$\mathbf{95 \pm 1}$	$\mathbf{95 \pm 1}$	79 ± 1	96 ± 0
R	14 ± 2	59 ± 4	68 ± 7	65 ± 4	$\mathbf{75 \pm 6}$	71 ± 1	92 ± 0
D	10 ± 0	31 ± 3	29 ± 2	$\mathbf{32 \pm 3}$	29 ± 5	$\mathbf{32 \pm 5}$	45 ± 2

for state-of-the-art methods on CARLA is to report only the evaluation variance by running multiple evaluations of a single training seed. We argue (given our findings) that for fair comparison, future studies should additionally report results by varying the training seed and providing the mean and standard deviation (as in Table II).

Comparison to State-of-the-Art: In our final experiment, we compare our approach to CAL [25], CILRS [6], LaTeS [34] and LSD [19], which is the state-of-the-art driving agent on the NoCrash benchmark with CARLA version 0.8.4. For fair comparison, we report percentage success rate with the mean and standard deviation over three different evaluations of the best model for each approach. We further report the results of the expert autopilot used for training on the CARLA simulator as an upper bound. Our results are summarized in Table III. We would additionally like to mention that LBC [4], which is the state-of-the-art for a different CARLA version (0.9.6) cannot be directly compared to these methods due to the reliance on several different forms of privileged information (such as the 3D position and orientation of all external dynamic agents).

Conditional Affordance Learning (CAL) [25], which maps an input image to six scalar 'affordances' that are used by a hand-designed controller for driving, is unable to achieve satisfactory performance. We rerun CILRS [6] using the author-provided best model, and notice that the rerun numbers (reported in Table III) differ significantly from the CILRS models we trained in our experiments (reported in Table II) despite using the same author-provided codebase. The authors do not release the specific dataset used for training their best model, which could explain the difference in performance. However, our models significantly outperform both the CILRS models we trained in our experiments (reported in Table II) and author-provided best model (reported in Table III) on every evaluation setting of the benchmark.

The authors of LaTeS [34] train a teacher network that takes ground truth segmentation masks as inputs and outputs low-level driving controls. A second student network which outputs driving controls by taking only RGB images as inputs is trained with an additional loss enforcing its latent

embeddings to match the teacher network. The training of their teacher network requires fine-grained semantic segmentation labels for each sample used to train the driving policy (hundreds of thousands of images). In contrast, the models trained on our Standard and Hybrid datasets require only a few hundred fine or coarsely labeled images respectively, and outperform LaTeS in the majority of the evaluation settings.

LSD [19] uses a mixture model trained using demonstrations and further refined by optimizing directly for the driving task in terms of a reward function. In contrast to our approach, this method uses no image-level annotations, but directly optimizing the policy with a reward is challenging outside of simulated environments. While this approach is slightly better at navigating empty conditions, our models outperform it in regular and dense traffic.

V. Conclusion

In this work, we take a step towards understanding how to efficiently leverage segmentation-based representations in order to learn robust driving policies in a cost-effective manner. As fine-grained semantic segmentation annotation is costly to obtain, and methods are often developed independently of the final driving task, we systematically quantify the impact of reducing annotation costs on a learned driving policy. Based on our experiments, we find that more detailed annotation does not necessarily improve actual driving performance. We show that with only a few hundred annotated images, that can be labeled in approximately 50 hours, segmentation-based visual abstractions can lead to significant improvements over end-to-end methods, in terms of both performance and variance with respect to different training seeds. Due to the modularity of this approach, its benefits can be extended to alternate policy learning techniques such as reinforcement learning. We believe that our findings will be useful to guide the development of better segmentation datasets and autonomous driving policies in the future.

Acknowledgements: This work was supported by the BMBF through the Tübingen AI Center (FKZ: 01IS18039B). The authors also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Kashyap Chitta and the Humboldt Foundation for supporting Eshed Ohn-Bar.

References

[1] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *arXiv.org*, 1604.07316, 2016. 2

[2] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 1 2009. 1, 2

[3] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 2

[4] D. Chen, B. Zhou, V. Koltun, and P. Krhenbhl. Learning by cheating. In *CoRL*, 2019. 7

[5] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, 2018. 2, 3

[6] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019. 2, 3, 4, 5, 7

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017. 2, 4

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8):1915–1929, 2013. 1

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012. 1

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[12] G. Heitz and D. Koller. Learning spatial context: using stuff to find things. In *ECCV*, 2008. 3

[13] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The ApolloScape Open Dataset for Autonomous Driving and its Application. *arXiv.org*, 1803.06184, 2018. 1, 2

[14] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 3

[17] A. Mousavian, A. Toshev, M. Fiser, J. Kosecká, A. Wahid, and J. Davidson. Visual representations for semantic target driven navigation. In *ICRA*, 2019. 1, 2

[18] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun. Driving policy transfer via modularity and abstraction. In *CoRL*, 2018. 1, 2

[19] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger. Learning situational driving. In *CVPR*, 2020. 2, 7, 8

[20] D. Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *NeurIPS*, 1988. 2

[21] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *CVPR*, 2020. 2

[22] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 7

[24] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *ECCV*, 2018. 3

[25] A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. In *CoRL*, 2018. 2, 7

[26] A. Sax, B. Emi, A. R. Zamir, L. J. Guibas, S. Savarese, and J. Malik. Learning to navigate using mid-level visual priors. In *CoRL*, 2019. 1, 2

[27] S. Shalev-Shwartz and A. Shashua. On the sample complexity of end-to-end training vs. semantic abstraction training. *arXiv.org*, 1604.06915, 2016. 1

[28] M. Toromanoff, E. Wirbel, and F. Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *CVPR*, 2020. 2

[29] D. Wang, C. Devin, Q. Cai, F. Yu, and T. Darrell. Deep object centric policies for autonomous driving. In *ICRA*, 2019. 2

[30] B. Wymann, C. Dimitrakakisy, A. Sumnery, E. Espié, and C. Guionneauz. Torcs: The open racing car simulator, 2015. 2

[31] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *CVPR*, 2017. 2

[32] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *arXiv.org*, 1805.04687, 2018. 1, 2

[33] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019. 2

[34] A. Zhao, T. He, Y. Liang, H. Huang, G. Van den Broeck, and S. Soatto. LaTeS: Latent Space Distillation for Teacher-Student Driving Policy Learning. *arXiv.org*, 1912.02973, 2019. 2, 5, 6, 7

[35] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2

[36] B. Zhou, P. Krähenbühl, and V. Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019. 1, 2

[37] A. Zlateski, R. Jaroensri, P. Sharma, and F. Durand. On the importance of label quality for semantic segmentation. In *CVPR*, 2018. 3, 5