SAM: Squeeze-and-Mimic Networks for Conditional Visual Driving Policy Learning

Albert Zhao^{*1} **Tong He**^{*1} **Yitao Liang**^{1}
Haibin Huang^{2} **Guy Van den Broeck**^{1} **Stefano Soatto**^{1}
^{1}University of California, Los Angeles, ^{2}Kuaishou Technology
{azzhao,simpleig,yliang,guyvdb,soatto}@cs.ucla.edu,
jackiehuanghaibin@gmail.com
* equal contribution

Abstract: We describe a policy learning approach to map visual inputs to driving controls conditioned on turning command that leverages side tasks on semantics and object affordances via a learned representation trained for driving. To learn this representation, we train a squeeze network to drive using annotations for the side task as input. This representation encodes the driving-relevant information associated with the side task while ideally throwing out side task-relevant but driving-irrelevant nuisances. We then train a mimic network to drive using only images as input and use the squeeze network's latent representation to supervise the mimic network via a mimicking loss. Notably, we do not aim to achieve the side task nor to learn features for it; instead, we aim to learn, via the mimicking loss, a representation of the side task annotations directly useful for driving. We test our approach using the CARLA simulator. In addition, we introduce a more challenging but realistic evaluation protocol that considers a run that reaches the destination successful only if it does not violate common traffic rules.

Keywords: autonomous driving, conditional imitation learning, side task

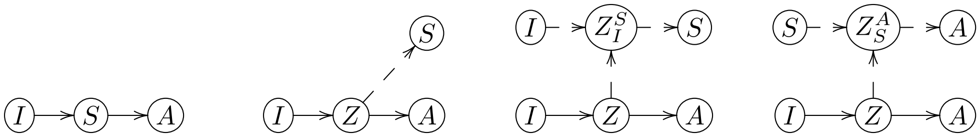
1 Introduction

Driving is a complex endeavor that consists of many sub-tasks such as lane following, making turns, and stopping for obstacles. A traditional strategy to tackle this complexity is to split the driving policy into two stages: perception (i.e. estimating a manually chosen representation of the scene) and control (i.e. outputting low-level controls using hand-coded rules). Though this approach is interpretable with easy-to-diagnose failures, it suffers from various issues: the representations may be suboptimal and difficult to estimate while hand-coded rules may struggle to handle the full range of driving scenarios [1, 2, 3]. These drawbacks motivate academia to explore learning approaches for driving as they are not restricted by hand-coded decisions. A major challenge for learning approaches is obtaining a useful representation. A simple approach is to directly learn a driving model that maps from image to low-level control. Though this approach is easy to implement, it suffers from poor generalization due to overfitting to nuisances [4, 5].

To improve generalization, various approaches (Fig. 1) propose to leverage complex side tasks such as semantic segmentation which provide driving-relevant contextual knowledge in an easily accessible form (i.e. without photometric nuisances). The most straightforward of these approaches is the two-stage approach [6, 7, 8, 9] (Fig. 1a), which learns to achieve the side task and then uses the output from the side task to estimate controls. However, this method suffers from errors in directly estimating the complex side task, leading to unrecoverable failures in the driving policy.

To avoid these estimation errors at test time, multi-task learning approaches [10, 4] (Fig. 1b) do not estimate the side task at test time but instead incorporate driving-relevant context by training a shared representation to achieve both the side task and the main driving task. However, they suffer from the fact that the side task is distinct from the main task (i.e. side task-main task mismatch), and hence, there exists information relevant to the side task but not to the main task. For example, the side task

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.



(a) Two-stage **(b)** Multi-task **(c)** Feature mimic **(d)** Ours

Fig. 1: Approaches for leveraging side tasks for driving. I, S, and A represent the image, side task annotations, and low-level controls; Z, Z_I^S, and Z_S^A, latent representation, mimicked features, and the squeezed encoding of the side task annotations. Dashed arrows, depending on direction, represent branches or supervision used only during training. Unlike other methods that aim to achieve the side task (two-stage, multi-task) or learn features for it (feature mimic), our method, SAM, learns only driving-relevant context from the side task annotations by directly feeding them as input during training. Also, note that SAM does not suffer from side task estimation errors encountered by other methods because we never explicitly estimate any side task annotations during training/inference.

of semantic segmentation encourages the learned representation to be informative for every pixel's class label, but knowing such pixel-wise information is unnecessary for driving. Instead, what is more important is knowing that an obstacle of a certain class is in front of the agent. Hence, as multi-task approaches aim to achieve both the side task and the driving task, they encourage the representation to contain side task-relevant but driving-irrelevant nuisances, hurting performance.

Feature mimicking [11] (Fig. 1c) incorporates contextual information while reducing the reliance on ground-truth annotations for the side tasks (side task annotations) by proposing instead to mimic features of a network (pre)trained to achieve the side task. However, this method suffers from a similar issue (side task-main task mismatch) to multi-task methods: as feature mimicking encourages the representation, via the mimicked features, to be informative for achieving the side task, it encourages the learning of side task relevant but driving-irrelevant nuisances. Moreover, all these previous methods (Fig. 1) inevitably suffer from side-task estimation errors during training/inference as they rely on training models to explicitly estimate the side task.

In contrast with previous approaches, we propose a method (Fig. 1d) that effectively leverages the side task for conditional driving policy learning without trying to achieve the side task or learn features trained to do so. Instead, we aim to learn a representation of the side task annotations that is directly useful for driving. We first train a squeeze network to drive using ground-truth side task annotations as input and extract its deep features. As these features are trained only for driving and not to achieve the side task, they contain only driving-relevant knowledge and not the side task-relevant but driving-irrelevant nuisances. However, we cannot use the squeeze network to drive as ground-truth side task annotations are unavailable at test time. To train a driving policy that does not require these annotations while still leveraging the side task, we train a mimic network to drive using only image and self-speed as input while pushing it to learn the squeeze network's embedding via a mimicking loss. Hence, the mimicking loss encourages the mimic network's representation to learn only the driving-relevant context associated with the side task. Our overall approach, squeeze-and-mimic networks (*SAM*), leverages the side task in an effective way to train a conditional driving policy without achieving the side task or learning features for it, avoiding the pitfalls of previous methods such as estimation errors and learning side task relevant but driving-irrelevant nuisances.

We note that for all the aforementioned methods (Fig. 1), the side tasks should not discard important driving-relevant information contained in the image input. Indeed, they should contain this information in an easily accessible form (i.e. without photometric nuisances) so that the intermediate representation can easily learn this information. Two important yet complementary types of information are object class, useful for tasks like lane following, and braking behavior. Hence, for our choice of side task annotations, we select semantic segmentation and *stop intention values,* which indicate whether to stop for different types of obstacles such as pedestrians, traffic lights, or cars.

We test our agent on online benchmarks using the photorealistic CARLA simulator [12]. Furthermore, we find previous benchmarks lacking as we do not think one should be rewarded for reaching the destination while driving through red lights or into the opposite lanes. Therefore we introduce a more realistic evaluation protocol that rewards successful routes only if they do not violate basic traffic rules. To summarize, our main contributions are as follows:

- a method for conditional driving policy learning that leverages a side task and learns only driving-relevant knowledge from its annotations, avoiding the pitfalls of existing methods that aim to achieve the side task or learn features for it.
- the combination of complementary side tasks: semantic segmentation, which provides basic class concepts for subtasks like lane following, and stop intentions, which provide causal information linking braking to hazardous driving scenarios. While both tasks lead to improvement over the baseline, we show that the best performance is achieved only when they are used jointly.
- an evaluation protocol, Traffic-school, that fixes the flaws of prior benchmarks. Though CARLA reports various driving infractions, these statistics are scattered and hard to analyze. Thus, previous benchmarks usually compute the route success rate, but they ignore several risky driving behaviors. In contrast, our protocol sets a high standard for the autonomous agent, penalizing previously ignored infractions such as violating red lights and running into the sidewalk.

Code is available at `https://github.com/twsq/sam-driving`.

2 Related Work

Autonomous driving has drawn significant attention for several decades [13, 14, 15]. Generally, there exist three different approaches to driving: modular, direct perception, and end-to-end learning.

Modular pipelines form the most popular approach [16, 17, 18, 19], separating driving into two components: perception modules [20, 21, 22] that estimate driving-related side tasks and control modules. Recently, [6, 7, 8, 9] use semantic segmentation and environment maps as side task annotations in a modular pipeline. However, these annotations are complex and high-dimensional. To simplify the annotations and alleviate the annotation burden, a recent work LEVA [8] proposes to use coarse segmentation masks. The mentioned modular approaches have the advantage of outputting interpretable representations in the form of estimated side task annotations (perception outputs), allowing for the diagnosis of failures. However, they suffer from perception and downstream control errors, *and* use manually chosen representations that are potentially suboptimal for driving.

Direct perception [23] methods similarly separate the driving model into two parts, but unlike modular approaches, they avoid complex representations, opting for compact intermediate representations instead. [24] and [3] (CAL) both train a network to estimate affordances (distance to vehicle, center-of-lane, etc.) linked to controls, for car racing and urban driving, respectively. Similar to modular approaches, the intermediate representation is hand-crafted, with no guarantees that it is optimal.

End-to-end methods [13, 14, 15, 25] learn a direct mapping from image to control and are most related to our work. CIL [26] builds upon offline behavioral cloning, solving the ambiguity problem at traffic intersections by conditioning on turning commands. [5] analyzes issues within the CIL approach and proposes an improved version, CILRS. However, both methods fail to generalize to dense traffic and suffer from the covariate shift between offline training data and closed-loop evaluation. To reduce this covariate shift, various methods collect data online using DAgger [27] imitation learning [28, 29] or reinforcement learning (RL) [30, 2, 31, 32]. Recently, [31] combines RL with a feature extractor pretrained on affordance estimation while LSD [32] combines RL with a multimodal driving agent trained via mixture of experts. However, the above methods all use online training, which is expensive and unsafe and can be performed only in simulation [12]. Due to these drawbacks, we focus on offline methods as offline and online methods are not directly comparable.

Previous methods have also leveraged side task annotations to improve generalization and acquire context knowledge. As discussed in the Introduction and shown in Fig. 1, the two-stage, multi-task and feature mimic methods [8, 10, 4, 11] all suffer from side task estimation errors and tend to learn driving-irrelevant nuisances (side task-main task mismatch). We overcome these issues by never aiming to achieve the side task; instead, we mimic a representation squeezed from side task annotations and trained for driving. Similar to our method, LBC [28] also mimics a driving agent. However, this agent, unlike ours, takes expensive 3D side task annotations as input. Moreover, LBC and SAM use different methods of mimicking: LBC mimics the agent's *controls* while SAM mimics the agent's *embeddings*. Hence, our method directly encourages the model's representation to contain driving-related context associated with the side task while LBC does not. To illustrate the benefits of our mimicking method, we also test SAM using LBC's control mimicking.

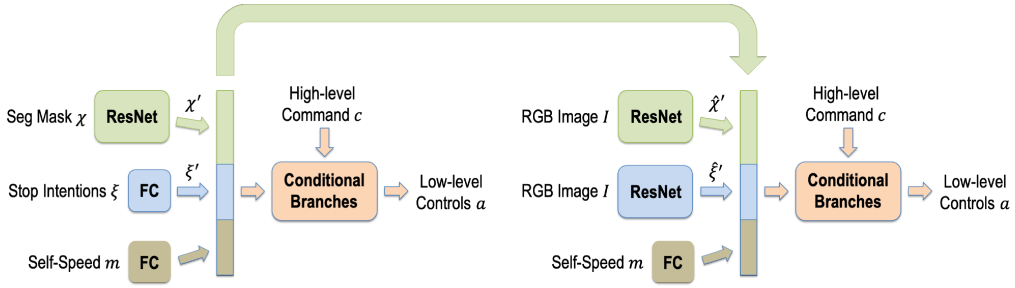


Fig. 2: Architecture overview of our squeeze and mimic networks. Both use a three-branch structure. (Left) Squeeze network: processes semantic segmentation, stop intention values, and self-speed in separate branches and feeds their concatenated representation to a command-conditioned driving module. (Right) Mimic network: has a similar structure but takes in only RGB image and self-speed. (Arrow) l_2 loss is enforced between the segmentation and intentions embeddings of the squeeze network and those of the ResNet branches, which each take in image, of the mimic network.

3 Method

The overall pipeline of our method *SAM* (squeeze-and-mimic networks) is shown in Fig. 2. The goal of conditional driving policy learning is to learn a policy $(I, m, c) \mapsto a$ that outputs low-level controls a given image I, self-speed m, and turning command $c = \{\text{follow}, \text{left}, \text{right}, \text{straight}\}$. Low-level continuous controls $a = (b, g, s)$ consist of brake b, gas g, and steering angle s.

For our method, we additionally leverage side task annotations $S = (\chi, \xi)$, in particular semantic segmentation χ and stop intention values ξ provided by the CARLA simulator, to train a conditional driving policy. The stop intentions $\xi = (v, p, l) \in [0, 1]^3$ indicate how urgent it is for the agent to brake in order to avoid hazardous traffic situations such as collision with vehicles, collision with pedestrians, and red light violations, respectively. They are like instructions given by a traffic-school instructor, which inform you of the causal relationships between braking and different types of dangerous driving scenarios, complementing the semantic segmentation χ, which contains concepts of object class identity for tasks like lane following. Hence, the training dataset for our method consists of temporal tuples $\{I_t, m_t, c_t, \chi_t, \xi_t, a_t\}_{t=1}^{T}$, collected from a rule-based autonomous driving agent with access to all internal state of the CARLA driving simulator.

In our method *SAM*, we first squeeze the side task annotations S_t, consisting of segmentation masks χ_t and three-category stop intentions $\xi_t = (v_t, p_t, l_t)$, into a latent representation $(\chi_t^{'}, \xi_t^{'})$ containing driving-relevant info and not driving-irrelevant nuisances by training a squeeze network to drive using side task annotations, self-speed, and turning command as input: $(S_t, m_t, c_t) \mapsto a_t$. We then train a mimic network $(I_t, m_t, c_t) \mapsto a_t$ to drive with no access to side task annotations while encouraging this network's embedding $(\hat{\chi}_t^{'}, \hat{\xi}_t^{'})$ to mimic the squeeze network's embedding. Notably, the squeeze network does not take image as input; this is so that its latent representation, which is used to supervise the mimic network, does not contain photometric nuisances from the image.

3.1 Squeeze Network

The task of our squeeze network is to squeeze the ground-truth segmentation masks χ_t and three-category stop intentions $\xi_t = (v_t, p_t, l_t)$ into a representation that contains contextual driving information but with nuisances removed. Such a representation will later be used to supervise the mimic network via a mimicking loss. As shown in Fig. 2, the squeeze network uses a three-branch architecture for estimating controls $a_t = (b_t, g_t, s_t)$. The first branch uses a ResNet34 backbone to squeeze the segmentation mask input, which provides concepts of object class useful for tasks like lane following. We use the middle fully-connected (FC) branch to process the three-category stop intentions, which complement the segmentation mask by informing the agent of the causal relationships between braking behaviors and the presence of objects in the context of safe driving. The lower FC branch ingests self-speed, which provides context for speeding up / slowing down [5].

The latent feature vectors from the three branches are concatenated and fed into a driving module, composed of several FC layers and a conditional switch [26] that chooses one out of four different output branches depending on the given turning signals c_t. The four output branches share the same network architecture but with separately learned weights. We use ℓ_1 losses for training:

$$L_{control} = \lambda_1 |\hat{b}_t - b_t| + \lambda_2 |\hat{g}_t - g_t| + \lambda_3 |\hat{s}_t - s_t| \tag{1}$$

where $(\hat{b}_t, \hat{g}_t, \hat{s}_t)$ are estimated controls of brake, throttle, steering angle respectively, and (b_t, g_t, s_t) are ground-truth controls. λ_i's are weights for loss terms.

3.2 Mimic Network

The mimic network does not have direct access to side task annotations, but instead observes a single RGB image I_t, self-speed measurement m_t, and high-level turning command c_t and mimics the squeeze network's latent embedding to learn driving-relevant context without learning driving-irrelevant nuisances. Its goal is also to estimate low-level controls a_t, for which it also adopts a three-branch network. The first and the second branches are now both ResNet34 backbones, pretrained on ImageNet, that separately take in image I_t and output latent embeddings $(\hat{\chi}_t^{'}, \hat{\xi}_t^{'})$. We adopt separate branches as this separation aids mimicking the embeddings of the segmentation masks, which are pixel-wise and provide object class identity, and of the intention values, which are global and provide causal links between braking and dangerous situations, respectively.

When training the mimic network, as illustrated in Fig. 2, besides applying the ℓ_1 control loss (Eq. (1)), we enforce ℓ_2 mimicking losses with weights $\lambda_i^{'}$ to encourage the mimic network's embeddings $(\hat{\chi}_t^{'}, \hat{\xi}_t^{'})$ to mimic the squeeze network's embeddings $(\chi_t^{'}, \xi_t^{'})$ of the side task annotations:

$$L_{mimic} = \lambda_1^{'} \left\| \hat{\chi}_t^{'} - \chi_t^{'} \right\|_2^2 + \lambda_2^{'} \left\| \hat{\xi}_t^{'} - \xi_t^{'} \right\|_2^2 . \tag{2}$$

The mimicking and control losses are combined to learn a driving policy: $L = L_{control} + L_{mimic}$.

3.3 Implementation Details

We implement our approach using CARLA 0.8.4 [12]. To train both the squeeze and mimic networks, we use a batch size of 120 and the ADAM optimizer with initial learning rate 2e-4. We use an adaptive learning rate schedule, which reduces the learning rate by a factor of 10 if the training loss has not decreased in 1000 iterations. We use a validation set for early stopping, validating every 20K iterations and stopping training when the validation loss stops decreasing.

Regarding time complexity, our mimic network demonstrates real-time performance (59 FPS) on a Nvidia GTX 1080Ti. Training the squeeze network takes 10 hours on a Nvidia GTX 1080Ti while the mimic network trains in 1 day on a Nvidia Titan Xp.

4 Experiments

We demonstrate the effectiveness and generalization ability of our method by evaluating it on standard closed-loop evaluation benchmarks. In section 4.1, we compare our model against a suite of competing approaches. Concerned about the fact that multiple driving infractions are not penalized in existing benchmarks, we then introduce a more realistic new evaluation standard *Traffic-school* in section 4.2. In addition, in sections 4.3 and 4.4, we conduct ablation studies on the strategy for leveraging the side task and individual types of side task annotations. For all benchmarks, we evaluate in four combinations of towns, which differ in road layout and visual nuisances, and weathers to test generalization: training conditions, new weather, new town, and new town/weather.

4.1 Comparison with State-of-the-Art on CARLA

We first test our method on the NoCrash and Traffic-light benchmarks [5]. We additionally present results on the saturated old CARLA [26] benchmark in Supp. Mat. For NoCrash, we also include, for completeness, results for LSD [32], LBC [28], and LEVA [8] even though they are not directly comparable to our method. LSD [32] uses online training, which is unsafe, and hence, is dependent on a driving simulator, while our method uses only offline training, avoiding these drawbacks. LBC [28] also uses online training and in addition uses expensive ground-truth 3D maps, collecting

	Training			New weather			New town			New town/weather			
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Mean
Comparable Baselines													
CIL [26]	79 ± 1	60 ± 1	21 ± 2	83 ± 2	55 ± 5	13 ± 4	48 ± 3	27 ± 1	10 ± 2	24 ± 1	13 ± 2	2 ± 0	36 ± 0.7
CAL [3]	81 ± 1	73 ± 2	42 ± 3	85 ± 2	68 ± 5	33 ± 2	36 ± 6	26 ± 2	9 ± 1	25 ± 3	14 ± 2	10 ± 0	42 ± 0.8
MT [4]	84 ± 1	54 ± 2	13 ± 4	58 ± 2	40 ± 6	7 ± 2	41 ± 3	22 ± 0	7 ± 1	57 ± 0	32 ± 2	14 ± 2	36 ± 0.8
Efficient seg* [8]	**100 ± 0**	82 ± 1	38 ± 5	80 ± 2	72 ± 3	23 ± 3	91 ± 1	58 ± 3	15 ± 4	68 ± 3	47 ± 5	19 ± 6	58 ± 1.0
Control mimic* [28]	**100 ± 0**	84 ± 1	31 ± 3	99 ± 1	76 ± 6	27 ± 6	84 ± 1	47 ± 2	11 ± 3	**83 ± 3**	51 ± 1	10 ± 2	59 ± 0.9
CILRS [5]	97 ± 2	83 ± 0	42 ± 2	96 ± 1	77 ± 1	39 ± 5	66 ± 2	49 ± 5	23 ± 1	66 ± 2	56 ± 2	24 ± 8	60 ± 1.0
SAM	**100 ± 0**	**94 ± 2**	**54 ± 3**	**100 ± 0**	**89 ± 3**	**47 ± 5**	**92 ± 1**	**74 ± 2**	**29 ± 3**	**83 ± 1**	**68 ± 7**	**29 ± 2**	**72 ± 0.9**
Listed for Completeness													
LSD [32]	N/A	N/A	N/A	N/A	N/A	N/A	94 ± 1	68 ± 2	30 ± 4	95 ± 1	65 ± 4	32 ± 3	N/A
LEVA [8]	N/A	N/A	N/A	N/A	N/A	N/A	87 ± 1	82 ± 1	41 ± 1	79 ± 1	71 ± 1	32 ± 5	N/A
LBC [28]	*100 ± 0*	*99 ± 1*	*95 ± 2*	*100 ± 0*	*99 ± 1*	*97 ± 2*	*100 ± 0*	*96 ± 5*	*89 ± 1*	*100 ± 2*	*94 ± 4*	*85 ± 1*	*96 ± 0.6*

Table 1: Results on NoCrash [5] benchmark. Empty, regular and dense refer to three levels of traffic. We show navigation success rate (%) in different test conditions. Due to simulator randomness, all methods are evaluated 3 times. Bold indicates best among comparable methods. Italics indicate best among all methods, including those whose results are provided solely for completeness such as LSD [32], LEVA [8], and LBC [28]. Note that LSD [32] and LEVA [8] report results only on the new town. We also include comparable baselines (marked with *), efficient seg and control mimic, for LEVA [8] and LBC [28]. The average error for *our model SAM* is **28%**, which is **30%** better than *the next-best CILRS*, **40%**, in terms of relative failure reduction.

	Train Town		New Town		
Method	Train weather	New weather	Train weather	New weather	Mean
CILRS [5]	59 ± 2	32 ± 1	43 ± 1	35 ± 2	42 ± 0.8
SAM	**97 ± 0**	**96 ± 1**	**81 ± 1**	**73 ± 1**	**87 ± 0.4**

Table 2: Traffic light success rate (percentage of not running the *red* light). We compare with the best comparable baseline CILRS.

which requires "accessing the ground-truth state of the environment," which "is difficult in the physical world" [28]. In contrast, our method uses only 2D segmentation and stop intentions. Finally, LEVA uses different pretraining (using MS-COCO segmentation dataset) and a significantly larger architecture (∼100M parameters for LEVA vs. ∼45M for our model). For fair comparisons, in Tab. 1, we also test LEVA (see "efficient seg") and LBC (see "control mimic") using comparable backbones and the same dataset as SAM. Details of these baselines are in Supp. Mat.

NoCrash We report results on the NoCrash benchmark in Tab. 1, where the metric is navigation success rate in three different levels of traffic: empty, regular, and dense. A route is considered successful for NoCrash only if the agent reaches the destination within the time window without crashing. Though the *second-best CILRS* outperforms the other comparable baselines, our method, with average error **28%**, still achieves a relative failure rate reduction of **30%** over CILRS, which has average error **40%**. In particular, our method achieves especially large performance gains over CILRS in new town and under regular and dense traffic. These gains demonstrate the effectiveness of both our choice of side task annotations (semantic segmentation for basic class concepts and stop intentions to aid braking) and our method of leveraging them. We also note that our method outperforms MT [4] (a multi-task method), control mimic, and efficient seg. This comparison illustrates the advantages of our method over other methods of using the side task annotations; our method learns only driving-relevant context associated with the side task. On the other hand, multi-task learning, efficient seg, and control mimic suffer from learning driving-irrelevant nuisances, perception errors, and not directly imbuing the representation with contextual knowledge, respectively.

Traffic-light Results are shown in Tab. 2. As NoCrash does not directly penalize running red lights, [5] proposes the Traffic-light benchmark to analyze traffic light violation behavior using the NoCrash empty setting (no dynamic agents). The metric is traffic-light success rate, i.e. the percentage of times the agent crosses a traffic-light on green. The traffic light success rate of *our model SAM* (**87%**) is more than twice as high as *CILRS* (**42%**). Our improved traffic light performance demonstrates the effectiveness of both our stop intentions and our method of leveraging them: the stop intentions inform the agent to stop for red lights while our method learns the context associated with them.

4.2 A More Realistic Evaluation Protocol: Traffic-school

To resolve the flaws of previous benchmarks, we propose the *Traffic-school* benchmark, which shares the same routes and weathers as NoCrash with a more restrictive evaluation protocol. In previous benchmarks, multiple driving infractions such as red light violations (ignored by NoCrash) and

	Training			New weather			New town			New town/weather			
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Mean
CILRS [5]	11 ± 2	12 ± 1	13 ± 2	2 ± 2	7 ± 2	7 ± 1	2 ± 1	7 ± 1	4 ± 1	0 ± 0	7 ± 1	2 ± 2	6 ± 0.4
SAM	$\mathbf{90 \pm 2}$	$\mathbf{79 \pm 1}$	$\mathbf{43 \pm 5}$	$\mathbf{83 \pm 3}$	$\mathbf{73 \pm 1}$	$\mathbf{39 \pm 4}$	$\mathbf{46 \pm 2}$	$\mathbf{39 \pm 3}$	$\mathbf{12 \pm 2}$	$\mathbf{15 \pm 3}$	$\mathbf{25 \pm 2}$	$\mathbf{14 \pm 0}$	$\mathbf{47 \pm 0.8}$

Table 3: The newly proposed ***Traffic-school*** benchmark provides a more solid evaluation standard than both the CARLA, Supp. Mat., and NoCrash, Tab. 1, benchmarks, which are flawed due to not penalizing infractions such as red light or out-of-road violations. On our new benchmark, a route is considered successful only if the agent arrives at the destination within a given time without a) crashing, b) traffic light violation, c) out-of-road infraction. Under this more realistic evaluation protocol, our results, in all conditions, surpass the best comparable baseline CILRS.

out-of-lane violations are ignored when judging whether a route is successfully finished. In the *Traffic-school* benchmark, we do not ignore such infractions; a route is considered a success only if the agent reaches the destination while satisfying the following requirements: a) no overtime, b) no crashes, c) no traffic light violation, d) no running into the opposite lane or sidewalk. As shown in Tab. 3, under this more realistic evaluation protocol, our results (**47%**) outperform the best comparable baseline CILRS (**6%**), showing that our method learns the driving-related context and not the nuisances associated with side tasks. In particular, effectively leveraging the semantic segmentation and stop intentions boosts our performance for staying in the lane and stopping for red lights (Tab. 2), infractions previously ignored but tested by *Traffic-school*.

4.3 Ablation Studies about Squeeze-and-Mimic Networks

To demonstrate the effectiveness of our method of using side task annotations, we conduct an ablation study on different methods of leveraging side tasks for driving (Tab. 4). We compare against alternative methods for driving policy learning [6, 9, 10, 4, 11], depicted in Fig. 1, using the same side tasks and comparable backbones. We also compare against baselines that do not use side tasks. In the Supp. Mat., we visualize saliency maps to qualitatively show the advantages of our method.

Two-stage-(F) We apply an intuitive strategy (Fig. 1a) of utilizing two separately trained modules: a) perception networks, b) driving networks. The perception networks are trained for segmentation masks and stop intention values estimation. In the second step, the driving networks use the same architecture as the squeeze network and take estimated segmentation masks and stop intentions as input for low-level controls estimation. For two-stage, we directly take the learned weights from the squeeze network as the driving network. Note that the squeeze network is trained with ground-truth segmentation masks and stop intention values. Thus, for two-stage-F, we fine-tune the driving network on the estimated segmentation masks and stop intentions. Using either variant, we note that this strategy suffers from perception errors, which cannot be recovered from.

Multi-task We apply a similar multi-task training strategy (Fig. 1b) as [10] and MT [4] but with our side tasks. On the same latent feature vectors where we enforce mimicking losses in our SAM method, we now train decoders to estimate segmentation masks and stop intentions as side tasks. The motivation is that by simultaneously supervising these side tasks, the learned features contain driving-relevant context such as lane markings and other agents and are more invariant to environmental changes like buildings, weather, etc. However, the downside of this side task supervision is that it encourages the learned features to contain side task-relevant but driving-irrelevant nuisances.

Feature mimic Inspired by [11], we construct a feature mimicking baseline (Fig. 1c) using our side tasks. Instead of mimicking the embeddings of a squeeze network trained to drive (SAM), we now mimic the embeddings of networks trained for semantic segmentation and stop intentions estimation. Similar to multi-task learning, feature mimicking also imbues the learned features with driving-relevant context but suffers from learning side task-relevant but driving-irrelevant nuisances.

No mimicking We also compare against two baselines that do not use side tasks, SAM-NM and Res101-NM, to analyze the impact of effectively leveraging the side tasks via our method. For both models, we simply use ℓ_1 losses for estimating low-level controls without enforcing the mimicking losses. SAM-NM uses the same two-ResNet architecture as SAM. As using two separate branches to process the image may be suboptimal in the no-mimicking case, we also compare to Res101-NM, a single-ResNet baseline that has a comparable number of network parameters.

From Tab. 4, we see that our method outperforms the alternative approaches leveraging the side task. It outperforms multi-task and feature mimic, showing that it effectively learns a representa-

	Training weather			New weather			
Method	Empty	Regular	Dense	Empty	Regular	Dense	Mean
SAM-NM	65 ± 3	36 ± 1	9 ± 2	42 ± 3	31 ± 2	7 ± 3	31.7 ± 1.0
Res101-NM	70 ± 3	44 ± 2	13 ± 4	50 ± 2	33 ± 1	7 ± 3	36.2 ± 1.1
Two-stage	**92 ± 1**	50 ± 3	12 ± 1	81 ± 2	41 ± 6	9 ± 3	47.5 ± 1.3
Two-stage-F	90 ± 2	57 ± 4	13 ± 1	79 ± 3	42 ± 4	8 ± 2	48.2 ± 1.2
Feature mimic	90 ± 1	62 ± 2	18 ± 1	79 ± 2	58 ± 2	15 ± 5	53.7 ± 1.0
Multi-task	91 ± 0	62 ± 2	17 ± 2	**83 ± 1**	65 ± 6	16 ± 2	55.7 ± 1.2
SAM	**92 ± 1**	**74 ± 2**	**29 ± 3**	**83 ± 1**	**68 ± 7**	**29 ± 2**	**62.5 ± 1.4**

Table 4: Comparison of alternative methods that leverage the side task on NoCrash. We show navigation success rate in the new town. Training town results are included in Supp. Mat. Though two-stage-(F), multi-task, and feature mimic improve over the aforementioned non-distillation models by large gaps, they still perform worse than our SAM model, showing that among multiple alternatives of using the segmentation masks and stop intention values, our method performs best.

	Training weather			New weather			
Mimicking source	Empty	Regular	Dense	Empty	Regular	Dense	Mean
Only stop intention	86 ± 2	47 ± 3	8 ± 4	73 ± 3	53 ± 6	9 ± 5	46.0 ± 1.7
Only seg. mask	**93 ± 1**	50 ± 4	8 ± 4	**85 ± 1**	52 ± 7	7 ± 3	49.2 ± 1.6
Both	92 ± 1	**74 ± 2**	**29 ± 3**	83 ± 1	**68 ± 7**	**29 ± 2**	**62.5 ± 1.4**

Table 5: Comparison of mimicking different types of knowledge. We show navigation success rate in the new town on NoCrash. Training town results are included in Supp. Mat. *Only stop intention* and *only segmentation mask* both improve upon the SAM-NM no-mimicking baseline, 31.7%, in Tab. 4. The best results are achieved by mimicking both types of embedding knowledge jointly.

tion that contains only the driving-relevant information associated with the side task and not the nuisances. In addition, it outperforms two-stage-(F), avoiding the perception errors that plague two-stage methods. Finally, we note that all methods using the side tasks outperform the no-mimicking baselines, showing that our choice of side tasks (segmentation masks, stop intentions) is effective for improving generalization and providing driving-relevant context.

4.4 Ablation Studies about the Chosen Side Task Annotations

We now conduct a careful ablation study to understand the impact of the chosen individual types of annotations. We analyze the influence of only utilizing one type of knowledge for mimicking: segmentation masks or stop intentions. In the Supp. Mat., we conduct ablation studies on the importance of each individual category of stop intention values, vehicle, pedestrian and traffic light, and show that the best performance is achieved when all three categories of intentions are used.

We use two different types of knowledge from the squeeze network for mimicking. Segmentation masks provide the mimic network with some simple concepts of object identities and therefore help the agent to learn basic driving skills like lane following and making turns. Meanwhile, stop intentions inform the agent of different hazardous traffic situations that require braking such as getting close to pedestrians, vehicles, or red lights. In Tab. 5 we conduct ablation studies mimicking each type of information separately. Both types of knowledge separately bring performance gains, but the best results are achieved only when they are used jointly due to their complementary nature.

5 Discussion

We propose squeeze-and-mimic networks, a method that encourages the driving model to learn only driving-relevant context associated with a side task while discarding nuisances. We accomplish this by first squeezing the complementary side task annotations, semantic segmentation and stop intentions, using a squeeze network trained to drive. We then mimic this network's representation to train the mimic network. Our method achieves state-of-the-art on various CARLA simulator benchmarks, including our newly proposed Traffic-school, which fixes previous benchmarks' flaws. In particular, *SAM* outperforms other approaches that also use side tasks.

However, our approach is not without limitations. Though it handles turning commands, it currently does not handle situations requiring negotiating with other agents such as lane changing and highway merging, a potential topic for future work. Sensor fusion with LiDAR to further improve dense traffic performance could be another interesting direction.

Acknowledgments

This work has been supported by grants ONR N00014-17-1-2072 and ARO W911NF-17-1-0304. This work has also been partially supported by NSF grants #IIS-1633857, #CCF-1837129, DARPA XAI grant #N66001-17-2-4032. We acknowledge the AWS Cloud Credits for Research program for providing computing resources.

References

[1] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *NeurIPS 2016 Learning, Inference and Control of Multi-Agent Systems Workshop*, 2016.

[2] X. Liang, T. Wang, L. Yang, and E. Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599, 2018.

[3] A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.

[4] Z. Li, T. Motoyoshi, K. Sasaki, T. Ogata, and S. Sugano. Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. *ArXiv*, abs/1809.11100, 2018.

[5] F. Codevilla, E. Santana, A. M. Lopez, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun. Driving policy transfer via modularity and abstraction. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1–15, 2018.

[7] J. Huang, S. Xie, J. Sun, Q. Ma, C. Liu, D. Lin, and B. Zhou. Learning a decision module by imitating driver's control behaviors. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[8] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger. Label efficient visual abstractions for autonomous driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[9] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems*, 2019.

[10] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538, 2016.

[11] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning to steer by mimicking features from heterogeneous auxiliary networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8433–8440, 2019.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1–16, 2017.

[13] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 305–313, 1988.

[14] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp. Off-road obstacle avoidance through end-to-end learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 739–746, 2005.

[15] D. Silver, J. A. Bagnell, and A. Stentz. Learning from demonstration for autonomous navigation in complex unstructured terrain. *International Journal of Robotics Research*, 29:1565–1592, 2010.

[16] S. Ullman. Against direct perception. *Behavioral and Brain Sciences*, 3:373–381, 1980.

[17] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. In *IEEE Transactions on Intelligent Vehicles*, volume 1, pages 33–55, 2016.

[18] U. Franke. *Autonomous Driving*, chapter 2, pages 24–54. John Wiley & Sons, 2017.

[19] E. Dickmanns and T. Christians. Relative 3d-state estimation for autonomous visual guidance of road vehicles. *Robotics and Autonomous Systems*, 7(2):113 – 123, 1991. Special Issue Intelligent Autonomous Systems.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[21] T. He and S. Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019.

[22] T. He, H. Huang, L. Yi, Y. Zhou, C. Wu, J. Wang, and S. Soatto. Geonet: Deep geodesic networks for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6888–6897, 2019.

[23] J. J. Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, 1979.

[24] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2722–2730, December 2015.

[25] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. M. adn Urs Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *ArXiv*, abs/1604.07316, 2016.

[26] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[27] S. Ross, G. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15, pages 627–635, 2011.

[28] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, 2019.

[29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11763–11773, 2020.

[30] Y. You, X. Pan, Z. Wang, and C. Lu. Virtual to real reinforcement learning for autonomous driving. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[31] M. Toromanoff, E. Wirbel, and F. Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7153–7162, 2020.

[32] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger. Learning situational driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11296–11305, 2020.

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[34] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, Jan 2018.

Supplementary Material

Contents

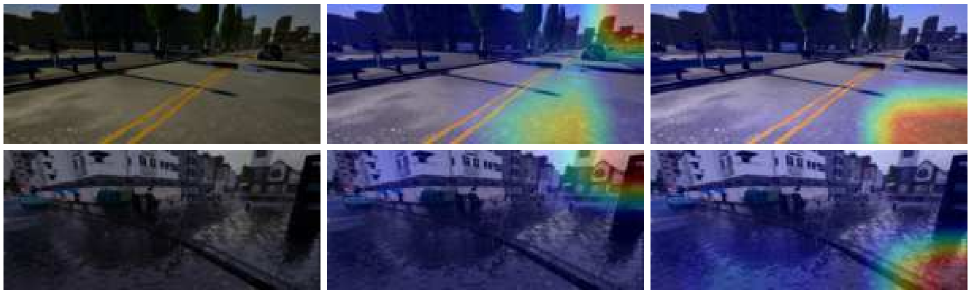


Fig. 3: Saliency heatmaps of the multi-task model (middle images) and our SAM model (right images) in the new town. The input images are on the left side. We see that our SAM model tends to focus on the road and sidewalk while the multi-task model tends to focus on driving-irrelevant nuisances such as the sky and buildings.

A Video Demo

Please see the Supp. Video (`https://youtu.be/VRlHJTxf0Uc`), which illustrates the diversity of conditions (weather, number of agents) as well as the covariate shift between the training town and the new town, with representative successful runs of various maneuvers (stopping for pedestrians, stopping at red light, turns with strong weather conditions, etc.). We also include a sample failure: colliding with vehicles while turning in dense traffic. We note that this failure mode appears rarely in the training dataset, and hence, our agent does not learn the proper behavior for this situation.

B Saliency Heatmaps

To show qualitatively that our method is less sensitive to driving-irrelevant nuisances, we visualize saliency heatmaps (Fig. 3), generated by GradCAM [33], for our SAM model and the multi-task model from our ablation study. We observe that our SAM model focuses its attention on driving-relevant entities like the road and sidewalk while the multi-task model focuses instead on driving-irrelevant nuisances such as the sky and buildings. These heatmaps, combined with the improved performance of our SAM method over multi-task learning (main paper Tab. 4), demonstrate that our method learns a representation containing only driving-relevant information associated with the side task unlike multi-task learning, which overfits to these nuisances leading to worse performance.

C Results on CARLA Benchmark

The CARLA benchmark [12] (Tab. 6) evaluates navigation with and without dynamic obstacles. The metric is navigation success rate, where a route is completed successfully if the agent reaches the destination within a time limit. This benchmark is relatively easy due to not penalizing crashes and therefore has been saturated. Nevertheless, *our model SAM* outperforms all comparable competing methods, achieving an average error of **2%** and a relative failure reduction of **85%** over the *second-best CILRS*, which achieves an average error of **13%**.

D Ablation Study about the Stop Intentions

An autonomous driving agent might push its brake for various reasons, such as approaching other vehicles, pedestrians, or a red light. To analyze the impact of individual stop intentions on the learned driving model, we present Tab. 7. The results indicate that when all three types of intentions are used, the agent achieves the best performance as the set of all three stop intentions provides a complete causal link between braking and different hazardous traffic situations for the evaluation benchmark we use. Combined with the ablation study about the segmentation mask and stop intentions annotations (main paper section 4.4), we see that it is beneficial to jointly mimic both three-category stop intention embeddings and segmentation mask embeddings to achieve the best performance.

Method	Training Static	 Dynamic	New weather Static	 Dynamic	New town Static	 Dynamic	New town/weather Static	 Dynamic	Mean
Comparable Baselines									
CIL [26]	86	83	84	82	40	38	44	42	62
CAL [3]	92	83	90	82	70	64	68	64	77
MT [4]	81	81	88	80	72	53	78	62	74
CILRS [5]	95	92	96	96	69	66	92	90	87
SAM	**100**	**100**	**100**	***100***	**95**	**92**	**98**	**98**	**98**
Listed for Completeness									
LSD [32]	N/A	N/A	N/A	N/A	99	98	*100*	98	N/A
LBC [28]	*100*	*100*	*100*	96	*100*	*99*	*100*	*100*	*99*

Table 6: Results on CARLA benchmark [12]. We show navigation success rate (%) in different test conditions. Dynamic / static indicate whether the test routes have moving objects (i.e. vehicles, pedestrians) or not. Bold indicates best among comparable methods. Italics indicate best among all methods, including methods such as LSD [32] and LBC [28] whose results are provided solely for completeness. Note that LSD [32] reports results only on the new town. *Our model SAM* achieves an average error of **2%**, surpassing the *second-best CILRS*, which has average error **13%**, by **85%** in terms of relative failure reduction.

Stop intention	Training Empty	 Regular	 Dense	New weather Empty	 Regular	 Dense	New town Empty	 Regular	 Dense	New town/weather Empty	 Regular	 Dense	Mean
Traffic light	95 ± 1	73 ± 1	16 ± 2	92 ± 0	63 ± 6	9 ± 1	60 ± 3	37 ± 4	11 ± 3	46 ± 2	25 ± 4	9 ± 1	45 ± 0.8
Vehicle	**100** ± **0**	89 ± 1	27 ± 3	94 ± 2	69 ± 1	25 ± 5	81 ± 2	50 ± 4	11 ± 2	73 ± 2	49 ± 7	13 ± 3	57 ± 0.9
Pedestrian	**100** ± **0**	93 ± 1	43 ± 2	98 ± 0	87 ± 5	41 ± 1	84 ± 2	61 ± 3	19 ± 1	71 ± 1	43 ± 1	13 ± 3	63 ± 0.6
All	**100** ± **0**	**94** ± **2**	**54** ± **3**	**100** ± **0**	**89** ± **3**	**47** ± **5**	**92** ± **1**	**74** ± **2**	**29** ± **3**	**83** ± **1**	**68** ± **7**	**29** ± **2**	**72** ± **0.9**

Table 7: Ablation study on three different categories of stop intention values, traffic light, vehicle and pedestrian, on NoCrash. We conduct ablation studies by mimicking the embeddings of squeeze networks trained using individual stop intentions. The best performance is achieved when all three-type stop intentions are used.

Furthermore, we observe that the only traffic light intentions model generally performs worse than the models using the other intentions. The relatively poor performance of this model could be due to the greater positive impact of the vehicle and pedestrian stop intentions during training compared to the traffic light intention. As the training data contains significant numbers of vehicles and pedestrians, the vehicle and pedestrian stop intentions may provide more benefit during training than the traffic light intention, leading to the performance gap between the traffic light intentions model and the other models.

E Additional Results on NoCrash Training Town

For the tables where only new town results were provided on NoCrash, here we demonstrate training town results. Tab. 8 and Tab. 9 provide the training town results corresponding to Tab. 4 and 5 in the main paper, respectively. We observe from Tab. 8, that our model outperforms the alternative methods of leveraging the side task in the training town, consistent with the new town results. We also see from Tab. 9 that mimicking both types of knowledge from the segmentation mask and stop intention values jointly provides the best performance in the training town, consistent with the new town results.

We now discuss an interesting observation from the ablation study for mimicking different types of knowledge (main paper Tab. 5). We observe that in the Empty setting, the model mimicking only segmentation mask embedding outperforms the model mimicking only stop intentions embedding. This trend can be explained by considering the different types of knowledge segmentation mask and stop intentions provide. Segmentation mask provides class knowledge for tasks such as lane following while stop intentions provide causal braking indicators for tasks such as collision avoidance. In the Empty setting, the agent does not need to stop for vehicles or pedestrians but still needs to follow the lane and make turns, so the braking knowledge provided by the stop intentions is not that important in this setting compared to the class knowledge provided by the segmentation mask. Hence, the only segmentation mask embedding model performs better in Empty compared to the only stop intentions embedding model.

Method	Training weather			New weather			Mean
	Empty	Regular	Dense	Empty	Regular	Dense	
SAM-NM	98 ± 0	81 ± 2	19 ± 3	96 ± 0	72 ± 4	18 ± 5	64.0 ± 1.2
Res101-NM	99 ± 1	85 ± 2	22 ± 1	89 ± 1	72 ± 3	27 ± 1	65.7 ± 0.7
Two-stage	**100 ± 0**	80 ± 4	29 ± 4	83 ± 1	63 ± 1	15 ± 4	61.7 ± 1.2
Two-stage-F	**100 ± 0**	83 ± 1	29 ± 4	87 ± 1	67 ± 2	22 ± 5	64.7 ± 1.1
Feature mimic	**100 ± 0**	87 ± 1	34 ± 3	**100 ± 0**	83 ± 5	34 ± 6	73.0 ± 1.4
Multi-task	**100 ± 0**	**94 ± 3**	41 ± 2	96 ± 0	87 ± 2	37 ± 5	75.8 ± 1.1
SAM	**100 ± 0**	**94 ± 2**	**54 ± 3**	**100 ± 0**	**89 ± 3**	**47 ± 5**	**80.7 ± 1.1**

Table 8: Comparison of alternative methods that leverage the side task on NoCrash [5] in the training town. We show navigation success rate (%) in different weather conditions.

Mimicking source	Training weather			New weather			Mean
	Empty	Regular	Dense	Empty	Regular	Dense	
Only stop intention	**100 ± 0**	86 ± 2	33 ± 4	97 ± 1	79 ± 1	25 ± 2	70.0 ± 0.8
Only seg. mask	**100 ± 0**	83 ± 3	31 ± 3	98 ± 2	79 ± 5	23 ± 6	69.0 ± 1.5
Both	**100 ± 0**	**94 ± 2**	**54 ± 3**	**100 ± 0**	**89 ± 3**	**47 ± 5**	**80.7 ± 1.1**

Table 9: Comparison of mimicking different types of knowledge on NoCrash in the training town. We show navigation success rate (%) in different weathers.

Method		Training			New weather			New town			New town/weather		Mean
	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
SAM-NM	32 ± 1	26 ± 2	10 ± 3	35 ± 3	21 ± 4	7 ± 4	9 ± 0	7 ± 1	3 ± 1	4 ± 0	9 ± 3	3 ± 2	14 ± 0.7
Res101-NM	35 ± 2	27 ± 2	15 ± 1	26 ± 2	16 ± 2	16 ± 2	14 ± 2	13 ± 3	6 ± 2	**17 ± 1**	14 ± 2	3 ± 1	17 ± 0.6
Two-stage	16 ± 0	12 ± 2	14 ± 1	9 ± 1	7 ± 1	4 ± 2	2 ± 1	9 ± 4	2 ± 1	5 ± 2	2 ± 0	2 ± 2	7 ± 0.5
Two-stage-F	19 ± 2	14 ± 1	18 ± 2	7 ± 1	8 ± 3	9 ± 2	3 ± 1	12 ± 3	4 ± 1	5 ± 2	1 ± 1	1 ± 2	8 ± 0.5
Feature mimic	22 ± 1	20 ± 1	19 ± 4	17 ± 1	17 ± 3	16 ± 3	11 ± 1	13 ± 1	8 ± 2	10 ± 0	15 ± 1	9 ± 3	15 ± 0.6
Multi-task	83 ± 1	72 ± 3	32 ± 1	62 ± 0	57 ± 3	28 ± 2	17 ± 1	16 ± 1	6 ± 0	9 ± 1	10 ± 2	5 ± 1	33 ± 0.5
SAM	**90 ± 2**	**79 ± 1**	**43 ± 5**	**83 ± 3**	**73 ± 1**	**39 ± 4**	**46 ± 2**	**39 ± 3**	**12 ± 2**	15 ± 3	**25 ± 2**	**14 ± 0**	**47 ± 0.8**

Table 10: Comparison of alternative methods that leverage the side task on Traffic-school. We show navigation success rate (%) in different test conditions.

F Additional Results on Traffic-school

We also provide results for the ablation studies on our newly proposed Traffic-school benchmark. Tab. 10, 11, 12 provide the Traffic-school results (both training and new towns) corresponding to main paper Tab. 4, 5, and Supp Mat Tab. 7, respectively. We see that on the Traffic-school benchmark, similar to the results on NoCrash, our model outperforms all alternative methods of leveraging the side task as it avoids perception errors (*e.g.* errors in segmentation mask and stop intentions estimation) that plague two-stage methods and learns only driving-relevant context squeezed by the squeeze network from the side task annotations unlike multi-task learning and feature mimicking. We also observe that both our method and multi-task learning outperform the no-mimicking methods on Traffic-school. This demonstrates the effectiveness of our choice of side tasks: semantic segmentation provides object class identity, aiding with basic tasks such as lane following, while three-category stop intentions provide causal information relating braking to various stopping causes. Furthermore, similar to the results on NoCrash, we observe that jointly mimicking both the segmentation mask and *three-category* stop intentions embeddings leads to the best performance on Traffic-school, demonstrating the complementary nature of the segmentation mask and three-category stop intentions side tasks.

G Comparison with Squeeze Network

We compare to the squeeze network in Tab. 13 and Tab. 14 to evaluate how well our SAM model (mimic network) mimics the squeeze network's embedding for the purpose of driving. We note that this comparison is not exactly fair as the squeeze network has access to ground-truth semantic segmentation and stop intention values at test-time unlike our SAM model, which has access to only images. Since we assume that the squeeze network's ground-truth inputs are generally unavailable at test time, for this comparison, we provide these inputs to the squeeze network.

As expected, the squeeze network, which has access to the ground-truth semantic segmentation and stop intention values at test time, generally outperforms the SAM model. However, the SAM model does not lag too far behind compared to the squeeze network (6% worse in success rate on NoCrash and comparable performance on Traffic-school), showing that the SAM model effectively mimics the squeeze network's embedding. The SAM model will occasionally outperform the squeeze net-

	Training			New weather			New town			New town/weather			Mean
Mimicking source	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
Only stop intention	28 ± 3	19 ± 2	22 ± 5	21 ± 1	17 ± 2	10 ± 3	7 ± 1	8 ± 2	2 ± 1	6 ± 2	11 ± 3	4 ± 0	13 ± 0.7
Only seg. mask	36 ± 3	28 ± 3	16 ± 3	23 ± 3	26 ± 0	11 ± 3	6 ± 1	10 ± 1	2 ± 1	7 ± 3	11 ± 5	5 ± 1	15 ± 0.8
Both	**90 ± 2**	**79 ± 1**	**43 ± 5**	**83 ± 3**	**73 ± 1**	**39 ± 4**	**46 ± 2**	**39 ± 3**	**12 ± 2**	**15 ± 3**	**25 ± 2**	**14 ± 0**	**47 ± 0.8**

Table 11: Comparison of mimicking different knowledge types on Traffic-school. We show navigation success rate (%) in different test conditions.

	Training			New weather			New town			New town/weather			Mean
Stop intention	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
Traffic light	34 ± 1	21 ± 3	11 ± 2	26 ± 0	10 ± 3	3 ± 3	9 ± 0	11 ± 3	5 ± 1	6 ± 2	9 ± 1	5 ± 1	13 ± 0.6
Vehicle	36 ± 1	30 ± 1	11 ± 2	20 ± 2	17 ± 3	5 ± 1	5 ± 1	7 ± 2	1 ± 2	5 ± 1	6 ± 2	4 ± 2	12 ± 0.5
Pedestrian	55 ± 1	51 ± 2	29 ± 1	54 ± 0	48 ± 5	23 ± 5	10 ± 2	10 ± 2	3 ± 2	7 ± 3	5 ± 1	2 ± 0	25 ± 0.7
All	**90 ± 2**	**79 ± 1**	**43 ± 5**	**83 ± 3**	**73 ± 1**	**39 ± 4**	**46 ± 2**	**39 ± 3**	**12 ± 2**	**15 ± 3**	**25 ± 2**	**14 ± 0**	**47 ± 0.8**

Table 12: Ablation study on three different categories of stop intention values on Traffic-school. We show navigation success rate (%) in different test conditions.

	Training			New weather			New town			New town/weather			Mean
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
SAM	**100 ± 0**	**94 ± 2**	54 ± 3	**100 ± 0**	89 ± 3	47 ± 5	92 ± 1	74 ± 2	29 ± 3	83 ± 1	68 ± 7	29 ± 2	72 ± 0.9
Squeeze	**100 ± 0**	93 ± 2	**63 ± 7**	**100 ± 0**	**93 ± 2**	**59 ± 4**	**97 ± 1**	**76 ± 3**	**40 ± 4**	**99 ± 2**	**81 ± 3**	**39 ± 1**	**78 ± 0.9**

Table 13: Comparison of SAM and squeeze network on the NoCrash benchmark. We show navigation success rate (%) in different test conditions.

	Training			New weather			New town			New town/weather			Mean
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
SAM	**90 ± 2**	**79 ± 1**	43 ± 5	**83 ± 3**	**73 ± 1**	39 ± 4	**46 ± 2**	39 ± 3	12 ± 2	15 ± 3	25 ± 2	14 ± 0	47 ± 0.8
Squeeze	76 ± 1	61 ± 1	**45 ± 4**	75 ± 2	61 ± 4	**45 ± 10**	39 ± 1	**40 ± 2**	**23 ± 4**	**39 ± 3**	**43 ± 1**	**23 ± 1**	**48 ± 1.1**

Table 14: Comparison of SAM and squeeze network on the Traffic-school benchmark. We show navigation success rate (%) in different test conditions.

	Training			New weather			New town			New town/weather			Mean
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	
SAM-NM (0 hours)	98 ± 0	81 ± 2	19 ± 3	96 ± 0	72 ± 4	18 ± 5	65 ± 3	36 ± 1	9 ± 2	42 ± 3	31 ± 2	7 ± 3	48 ± 0.8
SAM 5 hours	**100 ± 0**	**94 ± 1**	53 ± 3	**100 ± 0**	91 ± 1	43 ± 7	85 ± 2	61 ± 3	24 ± 6	76 ± 2	56 ± 9	21 ± 2	67 ± 1.2
SAM 7 hours	**100 ± 0**	93 ± 1	53 ± 5	99 ± 1	**92 ± 4**	43 ± 5	**94 ± 0**	72 ± 2	**31 ± 4**	**83 ± 1**	**69 ± 1**	27 ± 2	71 ± 0.8
SAM 10 hours	**100 ± 0**	**94 ± 2**	**54 ± 3**	**100 ± 0**	89 ± 3	**47 ± 5**	92 ± 1	**74 ± 2**	29 ± 3	**83 ± 1**	68 ± 7	**29 ± 2**	**72 ± 0.9**

Table 15: Comparison of SAM with differing amounts of side task annotations on NoCrash. We show navigation success rate (%) in different test conditions.

work due to overfitting to photometric nuisances in the training town. As a consequence, our SAM model does not generalize as well to the new town compared to the squeeze network, which has access to ground truth side task annotations.

H Ablation Study on Amount of Side Task Annotations

To examine the performance of our SAM method with varying amounts of side task annotations, we also train our SAM method using 0 hours (SAM-NM), 5 hours, and 7 hours of annotations (semantic segmentation and stop intentions) in addition to the default 10 hours. The results on NoCrash and Traffic-school are in Tab. 15 and 16, respectively. We observe that leveraging any amount (5 hours, 7 hours, or 10 hours) of side task annotations leads to significant performance gains over using no side task annotations, showing the effectiveness of our method even if side task annotations are not present for all frames, a situation that may occur in practice. Furthermore, we note that using 7 hours of side task annotations leads to comparable performance as using the full 10 hours of annotations, showing that our SAM method is robust to missing side task annotations in the dataset. Finally, we note that our SAM method (SAM 5 hours), despite using only half as many side task annotations, performs comparably to multi-task learning on NoCrash (67%, Tab. 15 vs. 66%, main paper Tab. 4 and Supp Mat Tab. 8) and outperforms multi-task learning on Traffic-school (43%, Tab. 16 vs. 33%, Tab. 10). As multi-task learning is the second-best method of leveraging the side tasks (main paper Tab. 4), this demonstrates that our method, among all the alternatives, most effectively leverages the side task as it learns a representation that contains driving-relevant context from the side task without the associated driving-irrelevant nuisances.

	Training			New weather			New town			New town/weather			
Method	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Mean
SAM-NM (0 hours)	32 ± 1	26 ± 2	10 ± 3	35 ± 3	21 ± 4	7 ± 4	9 ± 0	7 ± 1	3 ± 1	4 ± 0	9 ± 3	3 ± 2	14 ± 0.7
SAM 5 hours	84 ± 3	78 ± 1	**44 ± 1**	82 ± 3	**74 ± 9**	33 ± 6	40 ± 1	33 ± 1	12 ± 3	13 ± 1	13 ± 4	7 ± 1	43 ± 1.1
SAM 7 hours	**91 ± 1**	**82 ± 1**	42 ± 4	81 ± 1	**74 ± 5**	36 ± 7	42 ± 3	36 ± 1	**16 ± 2**	**23 ± 1**	**25 ± 2**	10 ± 2	**47 ± 0.9**
SAM 10 hours	90 ± 2	79 ± 1	43 ± 5	**83 ± 3**	73 ± 1	**39 ± 4**	**46 ± 2**	**39 ± 3**	12 ± 2	15 ± 3	**25 ± 2**	**14 ± 0**	**47 ± 0.8**

Table 16: Comparison of SAM with differing amounts of side task annotations on Traffic-school. We show navigation success rate $(\%)$ in different test conditions.

	Training			New weather			New town			New town/weather			
(λ_1', λ_2')	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Mean
(0, 0.03)	**100 ± 0**	86 ± 2	33 ± 4	97 ± 1	79 ± 1	25 ± 2	86 ± 2	47 ± 3	8 ± 4	73 ± 3	53 ± 6	9 ± 5	58 ± 0.9
(0.015, 0.03)	**100 ± 0**	**95 ± 1**	51 ± 3	**100 ± 0**	91 ± 5	43 ± 6	**95 ± 1**	68 ± 5	25 ± 6	75 ± 1	63 ± 1	23 ± 1	69 ± 1.0
(0.03, 0.03)	**100 ± 0**	94 ± 2	**54 ± 3**	**100 ± 0**	89 ± 3	**47 ± 5**	92 ± 1	**74 ± 2**	**29 ± 3**	83 ± 1	**68 ± 7**	**29 ± 2**	**72 ± 0.9**
(0.06, 0.03)	**100 ± 0**	94 ± 2	49 ± 4	**100 ± 0**	94 ± 5	45 ± 1	91 ± 1	70 ± 2	28 ± 5	63 ± 1	52 ± 5	23 ± 1	67 ± 0.8
(0.03, 0)	**100 ± 0**	83 ± 3	31 ± 3	98 ± 2	79 ± 5	23 ± 6	93 ± 1	50 ± 4	8 ± 4	**85 ± 1**	52 ± 7	7 ± 3	59 ± 1.1
(0.03, 0.015)	**100 ± 0**	**95 ± 2**	**54 ± 3**	**100 ± 0**	91 ± 1	44 ± 6	92 ± 1	71 ± 2	27 ± 3	82 ± 2	67 ± 2	22 ± 3	70 ± 0.8
(0.03, 0.06)	**100 ± 0**	94 ± 2	**54 ± 5**	**100 ± 0**	**95 ± 2**	**47 ± 6**	90 ± 1	70 ± 1	**29 ± 2**	62 ± 2	57 ± 9	16 ± 4	68 ± 1.1

Table 17: Ablation study on mimic loss weights (λ_1', λ_2') on NoCrash. We show navigation success rate $(\%)$ in different test conditions. We bold (λ_1', λ_2') = (0.03, 0.03) as this set of weights performs the best and is the one we use.

	Training			New weather			New town			New town/weather			
(λ_1', λ_2')	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Empty	Regular	Dense	Mean
(0, 0.03)	28 ± 3	19 ± 2	22 ± 5	21 ± 1	17 ± 2	10 ± 3	7 ± 1	8 ± 2	2 ± 1	6 ± 2	11 ± 3	4 ± 0	13 ± 0.7
(0.015, 0.03)	82 ± 2	77 ± 1	38 ± 2	75 ± 4	68 ± 5	37 ± 5	36 ± 1	25 ± 0	11 ± 4	14 ± 2	19 ± 1	9 ± 2	41 ± 0.8
(0.03, 0.03)	90 ± 2	79 ± 1	43 ± 5	**83 ± 3**	**73 ± 1**	39 ± 4	**46 ± 2**	39 ± 3	12 ± 2	15 ± 3	**25 ± 2**	**14 ± 0**	**47 ± 0.8**
(0.06, 0.03)	**91 ± 1**	**80 ± 3**	41 ± 1	77 ± 2	69 ± 3	**41 ± 3**	38 ± 2	35 ± 2	14 ± 3	**21 ± 1**	23 ± 3	11 ± 1	45 ± 0.7
(0.03, 0)	36 ± 3	28 ± 3	16 ± 3	23 ± 3	26 ± 0	11 ± 3	6 ± 1	10 ± 1	2 ± 1	7 ± 3	11 ± 5	5 ± 1	15 ± 0.8
(0.03, 0.015)	87 ± 1	**80 ± 2**	**45 ± 6**	76 ± 6	71 ± 4	33 ± 5	45 ± 1	**41 ± 4**	**17 ± 1**	18 ± 3	23 ± 5	11 ± 1	46 ± 1.1
(0.03, 0.06)	86 ± 1	78 ± 2	42 ± 2	73 ± 2	71 ± 3	38 ± 4	37 ± 2	37 ± 2	14 ± 1	12 ± 2	18 ± 9	9 ± 1	43 ± 1.0

Table 18: Ablation study on mimic loss weights (λ_1', λ_2') on Traffic-school. We show navigation success rate $(\%)$ in different test conditions. We bold (λ_1', λ_2') = (0.03, 0.03) as this set of weights performs the best and is the one we use.

I Ablation Study on Mimic Loss Hyperparameters

We conduct an ablation study (Tab. 17 and 18) on the mimic loss weights (λ_1', λ_2' in main paper equation 2) to examine the sensitivity of our SAM method with regards to these weights; note that our method uses $\lambda_1' = 0.03, \lambda_2' = 0.03$, the setting that performs best. We see that our method is fairly robust to various hyper-parameter settings as long as both mimic loss terms are used ($\lambda_1', \lambda_2' > 0$). Furthermore, we observe that performance decreases significantly with $\lambda_1' = 0$ or $\lambda_2' = 0$, showing that it is important to mimic both segmentation and stop intentions knowledge from the squeeze network.

J Detailed Model Architectures for Ablation Study

J.1 Our SAM Model

Network architectures of the squeeze network and the mimic network are explained in Tab. 19 and Tab. 20, respectively.

J.2 Two-stage-(F)

The two-stage-(F) model uses ErfNet [34] for semantic segmentation, which is also used in [6], and a ResNet34 based network, Tab. 21, for stop intentions estimation. The two-stage-(F) model then feeds the estimated segmentation masks and three-category intention values to a network with the same architecture as the squeeze network. In two-stage, we directly use the pretrained weights of the squeeze network, which is trained on ground-truth semantic segmentation and stop intention values. In two-stage-F, we further fine-tune the squeeze network on estimated perception inputs to reduce the impact of propagating estimation errors of the perception inputs.

Module	Layer	Input Dimension	Output Dimension
Segmentation Mask	ResNet34	88 × 200 × 6	512
Stop Intentions	FC	3 128	128 128
Self-Speed	FC	1 128	128 128
Joint Embedding	FC	512 + 128 + 128	512
Controls	FC	512 256 256	256 256 3

Table 19: Network architecture of the squeeze network. The dimension format is $height{\times}width{\times}channel$ for feature maps or just $channel$ for feature vectors.

Module	Layer	Input Dimension	Output Dimension
Seg Mask Embedding	ResNet34	88 × 200 × 3	512
Stop Intentions Embedding	ResNet34	88 × 200 × 3	128
Self-Speed	FC	1 128	128 128
Joint Embedding	FC	512 + 128 + 128	512
Controls	FC	512 256 256	256 256 3

Table 20: Network architecture of the mimic network. The dimension format is $height{\times}width{\times}channel$ for feature maps or just $channel$ for feature vectors.

J.3 Multi-task

For the multi-task model, we use the same architecture as the mimic network except we add two additional decoders that use the outputs of the segmentation mask embedding and stop intentions embedding branches to estimate segmentation mask and stop intentions, respectively. The architecture of the segmentation mask decoder is given in Tab. 22. This decoder consists of several deconvolution (*Deconv*) blocks (a deconvolution layer and a convolution layer), with skip connections between the outputs of the *Deconv* block and the outputs of the corresponding ResNet block from the ResNet34 encoder. The intentions decoder (Tab 23) has a similar architecture to the *Controls* module for the mimic network except that it takes in a vector of length 128, the length of the intentions embedding.

J.4 Feature mimic

For the feature mimic model, we train two networks for semantic segmentation and stop intentions estimation. Both networks use identical encoder architectures as the corresponding ResNet branches of the mimic network; this choice aids mimicking these networks' representations. For the decoders, we use the same decoder architectures as the multi-task model (Tab. 22 for segmentation mask decoder and Tab. 23 for intentions decoder).

J.5 Res101-NM

For the Res101-NM model, the architecture is the same as that of the mimic network except the segmentation mask and stop intentions embedding branches are replaced by a single ResNet101 branch with output FC vector size of $640 = 512 + 128$ to keep total latent embedding size the same as the mimic network.

K Efficient Seg and Control Mimic Baselines

We construct comparable baselines, efficient seg and control mimic, for LEVA [8] and LBC [28], respectively. Descriptions for both baselines are provided below.

Module	Layer	Input Dimension	Output Dimension
Perception	ResNet34	$88 \times 200 \times 3$	512
Stop Intentions	FC	512	256
		256	256
		256	3

Table 21: Network architecture of the stop intention estimation network used in Two-stage-(F).

Layer	Input Dimension	Output Dimension
FC	512	1536
Reshape	1536	$1 \times 3 \times 512$
Deconv (k2, s2, i512, o512, nb), BN, ReLU	$1 \times 3 \times 512$	$3 \times 7 \times 512$
Conv (k3, s1, i512, o512, nb), BN, ReLU	$3 \times 7 \times 512$	$3 \times 7 \times 512$
Deconv (k3, s2, i512, o256, nb), BN, ReLU	$3 \times 7 \times 512$	$6 \times 13 \times 256$
Conv (k3, s1, i256, o256, nb), BN, ReLU	$6 \times 13 \times 256$	$6 \times 13 \times 256$
Deconv (k3, s2, i256, o128, nb), BN, ReLU	$6 \times 13 \times 256$	$11 \times 25 \times 128$
Conv (k3, s1, i128, o128, nb), BN, ReLU	$11 \times 25 \times 128$	$11 \times 25 \times 128$
Deconv (k3, s2, i128, o64, nb), BN, ReLU	$11 \times 25 \times 128$	$22 \times 50 \times 64$
Conv (k3, s1, i64, o64, nb), BN, ReLU	$22 \times 50 \times 64$	$22 \times 50 \times 64$
Deconv (k3, s2, i64, o64, nb), BN, ReLU	$22 \times 50 \times 64$	$44 \times 100 \times 64$
Conv (k3, s1, i64, o64, nb), BN, ReLU	$44 \times 100 \times 64$	$44 \times 100 \times 64$
Deconv (k3, s2, i64, o64, nb), BN, ReLU	$44 \times 100 \times 64$	$88 \times 200 \times 64$
Conv (k3, s1, i64, o64, nb), BN, ReLU	$88 \times 200 \times 64$	$88 \times 200 \times 64$
Conv (k3, s1, i64, o6, nb)	$88 \times 200 \times 64$	$88 \times 200 \times 6$

Table 22: Network architecture of the segmentation mask decoder used in Multi-task and Feature mimic. Conv/Deconv (k, s, i, o, nb) denotes a Conv/DeConv layer with (kernel size, stride, input channels, output channels, no bias), and BN denotes a batch normalization layer.

Layer	Input Dimension	Output Dimension
FC	128	256
	256	256
	256	3

Table 23: Network architecture of the intentions decoder used in Multi-task and Feature mimic.

K.1 Efficient seg

Similar to LEVA, we construct a two-stage pipeline that uses coarse segmentation masks: we first estimate coarse segmentation masks and stop intentions and then train a driving network to drive using the estimated coarse masks and stop intentions.

Specifically, we train three networks:

1. we first train a network to estimate dense segmentation masks (the ErfNet [34] model used in two-stage-(F)). We then coarsen the estimated annotations for the *pedestrians*, *vehicles*, and *trafficSigns* classes using bounding boxes extracted from the estimated segmentation mask; LEVA coarsens the annotations for similar classes. We choose to estimate dense segmentation masks and then coarsen them as opposed to estimating coarse segmentation masks directly as we found that the first method yields better performance.

2. a network to estimate intentions (the same intentions estimation model used in two-stage-(F))

3. a model with the same architecture as the squeeze network trained to drive using the estimated coarse segmentation masks and stop intentions.

	Training			New weather			New town			New town/weather			
Method	Empty	Regular	Dense	Empty	Regular	Dense*	Empty	Regular	Dense	Empty*	Regular	Dense	Mean
CILRS Original	**97** ± **2**	**83** ± **0**	42 ± 2	96 ± 1	77 ± 1	**47** ± **5**	**66** ± **2**	49 ± 5	**23** ± **1**	**90** ± **2**	56 ± 2	24 ± 8	**63** ± **1.0**
CILRS Rerun	93 ± 1	**83** ± **2**	**43** ± **2**	**99** ± **1**	**81** ± **2**	39 ± 5	65 ± 2	**51** ± **1**	**23** ± **1**	66 ± 2	**59** ± **2**	**27** ± **3**	61 ± 0.7

Table 24: Comparison on NoCrash as reported in the *original* CILRS paper [5] v.s. our *rerun* with author-released code and model. We show navigation success rate $(\%)$ in different test conditions. Columns with * indicate evaluation settings where we report numbers from the *rerun* since the success rate differences are larger than 5%; otherwise, we report numbers from the *original* CILRS paper.

Method	Training	New weather	New town	New town/weather	Mean
CILRS Original	53	N/A	N/A	36	45
CILRS Rerun	59 ± 2	32 ± 1	43 ± 1	35 ± 2	42 ± 0.8

Table 25: Comparison on traffic light success rate (percentage of not running the *red* light) as reported in the *original* CILRS paper [5] v.s. our *rerun* using author-released code and model. We note that the CILRS paper does not report standard deviations, and results on new weather as well as new town conditions. In general, the *original* numbers are comparable with our *rerun* results.

K.2 Control mimic

Similar to LBC, we train the mimic network to mimic the privileged agent's (squeeze network's) controls; note that our SAM method instead mimics the squeeze network's embeddings. The control loss used is the same as used in our method except that we use the squeeze network's estimated controls on the training set as ground-truth controls instead. We also adopt the "whiteboxing" data augmentation technique used in LBC: for all frames in the training set, we mimic the squeeze network's estimated controls for all four high-level turning commands $c = \{\mathrm{follow}, \mathrm{left}, \mathrm{right}, \mathrm{straight}\}$, as opposed to only the turning command associated with each frame.

L Training Dataset

We collect a 10 hours ($\sim$ 360K frames from the front view, 10 FPS) training dataset using the same data collector as [5]. The dataset is collected in the training town (Town01) and four training weathers ("Clear Noon", "Heavy Rain Noon", "Clear Noon After Rain", and "Clear Sunset"), identical to the training conditions for all the evaluation benchmarks. For each episode, we randomly sample the number of vehicles and pedestrians from the ranges [30, 60] and [50, 100], respectively.

M Semantic Segmentation

For the semantic segmentation annotations, we retain classes relevant to driving and throw out nuisance classes. Specifically, we use the *pedestrians*, *roads*, *vehicles*, and *trafficSigns* classes and map the *roadlines* and *sidewalks* classes to the same class. We map all other classes to a nuisance class. Hence, we obtain a total of 6 classes for the semantic segmentation.

N CILRS Original v.s. Rerun

We rerun CILRS [5] using the author-provided code and model. Comparisons between the *original* and the *rerun* results are shown in Tab. 24 and Tab. 25. We notice that some of the *rerun* numbers differed significantly (by more than 5%) from those reported in the original CILRS paper. For these numbers, we report the numbers we obtained from rerunning their released code and model.

O Q&A

In this section, we address hypothetical questions that may arise from this work.

1. Why do you provide comparable baselines for LEVA [8] and LBC [28] but not LSD [32]?

We provide comparable baselines for LEVA [8] and LBC [28] so that we can compare our method of leveraging side tasks to their methods of leveraging side tasks on a fair basis (comparable backbones and same dataset). In contrast, for LSD [32], such a comparison is moot as LSD does not focus on leveraging side tasks but instead combines reinforcement learning with a multimodal mixture of experts driving model, an orthogonal direction to our work. For the best performance, one can integrate leveraging side tasks with an ensemble of driving models and reinforcement learning, but this is out of the scope of our work.

2. What are the advantages of Traffic-school over reporting separate metrics for different types of infractions?

Though CARLA reports various driving infractions, these statistics are scattered and hard to analyze, leading to prior works computing different infractions metrics that are not directly comparable. For example, Codevilla et al. [5] reports percentage of crossing red traffic lights while Chen et al. [28] reports number of traffic light violations per 10 km. To resolve this issue, previous benchmarks usually compute the route success rate, but they ignore several infraction types. We resolve both issues (scattered statistics and ignoring various infractions) by proposing Traffic-school, which computes route success rate while penalizing many basic infraction types (crashes, red light violations, out of lane violations) as failures.

3. Why do the results on Traffic-school not monotonically decrease as we go from Empty to Regular to Dense? For example, in main paper Tab. 3, CILRS success rates in new town are $2\%, 7\%, 4\%$ in Empty, Regular, Dense, respectively.

The results on Traffic-school do not necessarily decrease monotonically because the hand-coded non-player vehicles in the CARLA simulator stop for red lights. Hence, for agents that are better at stopping for vehicles than for red lights, stopping for vehicles may help with not violating red lights, leading to the performance on Traffic-school potentially increasing with a higher number of vehicles. This phenomenon ideally should not happen for agents that perform well in stopping for red lights, emphasizing the need for benchmarks such as Traffic-school that take red light violations into account.

4. Why does the SAM model perform better for New town/weather than for New town on the CARLA benchmark (Supp Mat Tab. 6)?

Generally, it would be expected that New town/weather would be the hardest setting. However, previous papers [12, 5] have also observed this same trend for the CARLA benchmark. Since many risky driving infractions (e.g. running into opposite lane, collision, etc.) are not penalized in this saturated and flawed old benchmark, the improved metrics of New town/weather does not indicate that the agent is actually performing better under this setting. This unexpected behavior is not observed in our newly proposed Traffic-school benchmark thanks to more realistic evaluation protocols than the old benchmarks.