Domain Adaptation Through Task Distillation

Brady Zhou*[0000−0001−7689−1960], Nimit Kalra*[0000−0001−9946−4945], and Philipp Krähenbühl[0000−0002−9846−4369]

The University of Texas at Austin, Austin TX, USA
{bzhou, nimit, philkr}@cs.utexas.edu

Abstract. Deep networks devour millions of precisely annotated images to build their complex and powerful representations. Unfortunately, tasks like autonomous driving have virtually no real-world training data. Repeatedly crashing a car into a tree is simply too expensive. The commonly prescribed solution is simple: learn a representation in simulation and transfer it to the real world. However, this transfer is challenging since simulated and real-world visual experiences vary dramatically. Our core observation is that for certain tasks, such as image recognition, datasets are plentiful. They exist in any interesting domain, simulated or real, and are easy to label and extend. We use these recognition datasets to link up a source and target domain to transfer models between them in a task distillation framework. Our method can successfully transfer navigation policies between drastically different simulators: ViZDoom, SuperTuxKart, and CARLA. Furthermore, it shows promising results on standard domain adaptation benchmarks.

Keywords: Domain Adaptation, Autonomous Driving, Sim-to-Real

1 Introduction

Labeled data has been the main driving force behind the rise of deep learning [7]. Tasks with an abundance of labeled data flourished [3,6,7,14,29], whereas tasks short on data saw only limited progress [31,45]. Over the past decade, deep networks have cut the error rate for image recognition by a factor of four [16,26,40], doubled object detection performance [13,15,29], and allow for a near-perfect pixel-wise segmentation of an image [5,58]. Yet, these same networks do not yet drive real-world autonomous vehicles, pilot a drone, or control a robot from the same diverse real-world visual inputs [3]. In simulation [10,24,43], these tasks are not necessarily much harder to learn than recognition [4] — they simply have little to no labeled real-world data. Unfortunately, models born and raised purely in simulation often fail to perform well in the real world [41,42]. This problem is not unique to simulated and real domains — even models trained on one specific dataset often fail to generalize to other datasets [51].

Our core observation is that the gap between many datasets is much smaller in the output labels than the input images, as shown in Figure 1. This is no

* indicates equal contribution

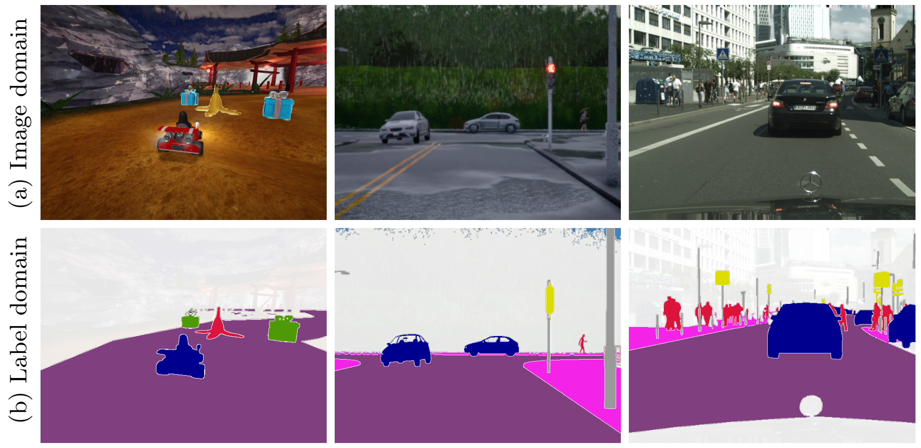


Fig. 1: Raw visual inputs (a) may significantly vary across different domains, yet they often share common recognition labels (b). In this work, we use these recognition labels to transfer tasks between different domains.

accident. Recognition tasks are carefully hand-designed to infer a compact, general, and abstract representation of the world [7, 12, 27, 29, 47]. Datasets often share largely overlapping label sets, and task definitions are compatible. Most recognition tasks were designed as a first stepping stone to the rich world of visual reasoning tasks [57, 59]. Why not solve recognition in all domains, and then build downstream tasks on top of recognition models [2, 33, 49, 56]? Since the gap between label spaces is small, representations will generalize. As it turns out, "solving recognition" turns out to be quite hard [3,11,29,40]. Current recognition systems mislabel objects, detect an object where there is none, or worse, fail to recognize objects altogether. If recognition is not solved in its entirety, errors will compound to downstream tasks, and a domain gap will persist between domains with good recognition systems and those with poor ones.

In this paper, we take a different approach. We use the ground truth recognition labels directly to transfer downstream tasks from a source to target domain through task distillation. First, we learn a proxy model that maps ground-truth recognition labels to outputs of the source model, through distillation [18]. This proxy model generalizes much better to the target domain, as it operates on a more compact and abstract input. Next, we perform a second step of distillation to recover an image-based target model that imitates the proxy. This target model, no longer uses any ground truth supervision and learns the task in an end-to-end manner. We call this procedure *task distillation* as it distills a source task to operate in a target domain with the help of an auxiliary recognition task.

This procedure may seem counterintuitive, but it has several advantages over other domain adaptation methods. Firstly, recognition labels from different domains exhibit a smaller domain shift than their raw image counterparts. Secondly, we do not need to solve recognition in either source or target domain

— we simply need a recognition dataset in each domain with a compatible label space. Finally, task distillation results in an end-to-end model in the target domain, and does not suffer from compounding errors in deployment.

We investigate how our task transfer framework performs under two distinct domain adaptation applications: 1) driving policy transfer for visual navigation, and 2) simulation-to-reality transfer for semantic segmentation prediction. We first show that our framework is able to transfer a lane-following driving policy from a simple racing game to a fully-fledged driving simulator. Furthermore, our framework is even able to transfer an obstacle-avoidance policy from a maze-navigation video game to driving among other moving vehicles. In the target domain, both of our transferred policies drive twice as far as the closest baselines.

Next, we apply our proposed framework to the standard domain adaptation task of transferring semantic segmentation models from simulated to real-world datasets. Here, task distillation again significantly outperforms prior work. Our framework is conceptually simple and easy to implement. All code and data is publicly available at https://github.com/bradyz/task-distillation.

2 Related Work

Dataset bias is likely as old as machine learning itself [51] — models trained on one dataset tend to generalize poorly out-of-the-box to related ones. The rise of deep learning ushered in a wave of massive datasets [3, 6, 7, 14, 29]. While these diverse datasets have significantly improved the general proclivity of state-of-the-art models to generalize to other datasets, a domain gap still exists. One popular and direct method to close this domain gap is to pre-train on a source domain and then fine-tune on a target domain [21, 30]. In the same spirit, our work leverages a large amount of ground-truth labels to provide supervision for vision tasks in a different domain. However, rather than rely on final task labels in the target domain, as they may be difficult or impossible to collect, we use generic recognition labels in both domains.

Domain adaptation aims to bridge the source and target domain by adapting the weights of a model to increase its performance in a target domain. Domain adaptation techniques include domain-specific normalization techniques [28], statistical matching on input [34, 53, 60], output [19, 52], and intermediate activation [20] distributions between source and target domains. While most techniques rely only on the statistical distributions of the input data and the transferred model, recent works have introduced auxiliary labels and tasks to aid adaptation [48, 54]. Ramirez et al. [37] learn a common representation for an auxiliary task in both the source and target domain, then use this representation to link the two domains and aid transfer between them. Our approach is significantly simpler and does not require training any models for the auxiliary task.

Simulation-to-reality transfer has received a lot of attention in recent years, as simulators effortlessly produce massive amounts of labeled data [25,38]. Transfer via modular pipelines is a promising simulation-to-reality method, wherein the observation space in both domains is mapped to a shared intermediate proxy task to ease generalization. Doersch et al. [8] use motion to transfer 3D human pose labels. Müller et al. [33] use a semantic segmentation proxy task to transfer a driving policy, whereas Mousavian et al. [32] use it as input to learned visual navigation policies. Zhou et al. [59] explore the impact of various high-level intermediate visual representations on learning to act, whereas Sax et al. [44] explore how mid-level visual priors assist learning to navigate.

While modular systems all greatly outperform using unsupervised domain adaptation techniques, they still suffer from compounding errors. If the intermediate auxiliary tasks is not solved perfectly, different error patterns between source and target domain will persist and lead to weaker transfer. Our framework, on the other hand directly uses the error-free ground truth annotations for transfer, and thus does not suffer from compounding errors.

Another popular avenue for simulation-to-reality transfer is through the use of domain randomization. In an effort to capture and generalize to the target domain distribution, these techniques randomize visual and dynamical properties of the simulation during training, similar to data augmentation in general deep learning. Domain randomization requires no real-world labels and are especially popular in transferring robotic manipulation tasks [1, 22, 35, 50]. James et al. successfully use this technique to reduce the amount of real world training data for robot grasping by two orders of magnitude [22]. While domain randomization works well in simple closed environments, it is not yet clear how to randomize more complex simulators to enable transfer to real-world visual domains.

3 Method

Our framework makes heavy use of model distillation [18]. Let $f_\theta(x)$ be a deep network with parameters θ. Let $g(x)$ be a second function, possibly another deep network. Distillation trains f_θ to produce the same output as g on a dataset D:

$$\text{minimize}_\theta E_{x \sim D} \left[\ell(f_\theta(x), g(x)) \right]$$

Distillation learns to imitate more than just the ground truth labels. It gets to see and imitate all outputs from a target function g, and thus captures some of its inner workings and representation, also known as dark knowledge [18]. The original distillation work [18] learns classification tasks using a smooth cross entropy loss ℓ. However, a large family of loss functions generally work. For simplicity, we use an L_1 loss $\ell(f_\theta(x), g(x)) = |f_\theta(x) - g(x)|$ for both categorical and regression tasks.

Distillation is easily extended to a pair of models that use different inputs x and y, as long as there exists a dataset with paired input modalities (x, y).

$$\text{minimize}_\theta E_{(x,y) \sim D} \left[\ell(f_\theta(x), g(y)) \right]$$

For notational simplicity, we call this process $f := \mathcal{D}_D(g)$ in later sections. Distillation is the cornerstone of our domain adaptation algorithm, and the generalized process is also pervasive in policy optimization under the term behavior

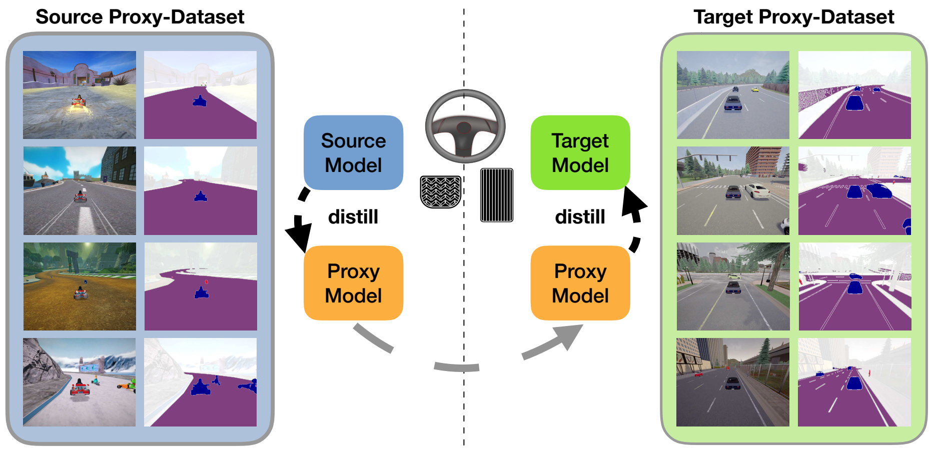


Fig. 2: Our method first distills a source model to a proxy model that uses labels as inputs. As proxy labels generalize to the target domain, a second stage of distillation is performed to produce a target model.

cloning or imitation learning [36]. In previous works, it has been successfully used to replace a privileged driving policy with a pure sensorimotor policy [4].

3.1 Task Distillation

Let $\mathcal{S}$ be the source domain and $\mathcal{T}$ be the target domain. We denote images from source and target domains as I^S and I^T, respectively. Let L^S and L^T be labels for a proxy task in both domains (e.g., image recognition). Our goal is to transfer a model $f^S : I^S \to o$ producing outputs o from the source domain to the target domain. In our formulation, this output represents the desired prediction we aim to transfer to the target domain; hence, we assume o is only available in the source domain. We want to learn an adapted model $f^T : I^T \to o$ that performs the same task as f^S in the target domain.

We propose a two-stage approach, as shown in Figure 2. First, we learn a proxy model $f^P : L^S \to o$ using model distillation on a source dataset; i.e: $f^P := \mathcal{D}_{\mathcal{S}}(f^S)$. Next, we distill the target model $f^T : I^T \to o$ from the proxy model on a target dataset, yielding $f^T := \mathcal{D}_{\mathcal{T}}(f^P)$. If the label sets do not perfectly align between source and target domains, we bring them closer using simple hand designed transformations, such as remapping semantic labels, or different forms of data augmentation, as described in Section 4.

3.2 Comparison with Modular Approach

Under which conditions does task distillation confer benefits over previous domain adaptation approaches? Which scenarios will likely cause it to fail? Both task distillation and the modular approach derive stronger generalization in the target domain through the use of an abstract, yet rich and informative, proxy task and proxy model. However, whereas task distillation queries the proxy model at train-time with *ground-truth* proxy labels, a modular pipeline queries this proxy model at deploy-time with *predicted* proxy labels, thereby incurring the cost of an imperfect recognition system.

We denote the accuracy of the recognition system in the target domain as a^l, the proxy model accuracy in the source domain as a^P, and the *similarity* between the two domains as $G^I = |I^S \cap I^T|/|I^T|$ in image space and $G^L = |L^S \cap L^T|/|L^T|$ in label space. Suppose a failure at any stage causes the entire system to fail. Then, the final accuracy is

$$a^T_{\text{modular}} = a^P a^l G^L; \tag{1}$$

that is, the system succeeds only if proxy, recognition, and label transfer succeed.

Similarly, let a^d be the accuracy of the second distillation stage in our proposed framework. Then, task distillation succeeds at a rate of

$$a^T_{\text{distill}} = a^P G^L a^d; \tag{2}$$

that is, when the labels transfer and the second distillation succeeds.

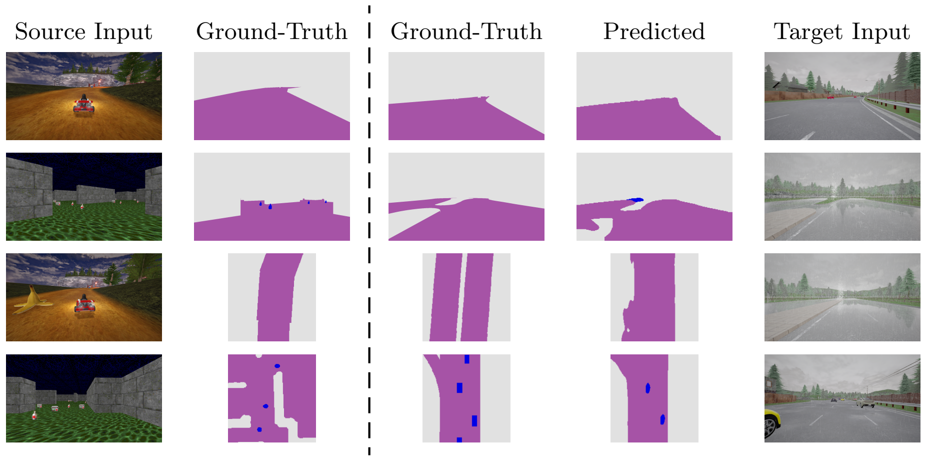
Finally, direct transfer ignores the issue of domain shift altogether, and simply evaluates the source model f^S in the target image domain. If f^S has accuracy a^S, this approach succeeds at a rate of

$$a^T_{\text{direct}} = a^S G^I. \tag{3}$$

These accuracy estimations can be difficult to compare without references or experiments. However, note that experimentally distillation commonly does not lose any accuracy — Hinton et al. [18] observe an increase in accuracy through distillation, while Chen et al. [4] show equivalent accuracies. It is thus safe to assume that $a^S \approx a^P$ in most cases.

With these estimates, we can reason about the relationships between domain adaptation approaches and develop some intuition. Firstly, if the domain gap in the image domain is not significantly larger than the domain gap in the label domain ($G^I \approx G^L$), then direct transfer is likely to work quite well. Moreover, task distillation and the modular approach differ by a single term — the recognition accuracy a^l versus the distillation accuracy a^d. If recognition is easier to learn in the target domain, a modular approach likely yields a higher accuracy. However, if the target task is easier to learn through distillation, a task distillation approach likely works better.

We find that, for many applications, modular pipelines must infer a proxy recognition task that is often far richer than needed for the end task. Target tasks are empirically easier to learn than recognition, as they can often be inferred from a subset of recognition. Although recognition is easier to supervise, with ground-truth labels in abundance, solving recognition pixel-perfect is very difficult. In contrast, the target task of driving from pixels, for example, only relies on inferring a subset of recognition. Intuitively, imperfect recognition 100m



(a) Source Domain (b) Target Domain

Fig. 3: We compare visual domains by their raw monocular images and corresponding semantic representations. While the domains vary significantly in their raw images, they are quite similar in their semantic modalities. However, note that the predicted modalities used by a modular pipeline are not perfect. For example, in the bottom-most row, the map-view prediction fails to capture the yellow car in view directly left of the agent. When supplied to the downstream driving policy, this vision failure can result in unintended behavior.

down the road, or on the opposite lane of a separated freeway, does not impact downstream driving performance.

We note that our task distillation framework and the modular approach both rely on a^P and G^L — their success depends on the performance of the proxy model in solving the final task and the ability of the proxy task to generalize between domains. Hence, the choice of proxy task is a careful trade-off between expressiveness and abstractness.

Although this analysis provides some intuition, our assumptions do ignore several practical aspects of modular approaches and distillation. Firstly, not all mistakes are equally bad — models are generally able to recover from partial failures and degrade slowly. Moreover, while the second stage of our task distillation framework may introduce some mislabeled training data if the label spaces are too dissimilar, deep learning algorithms (namely stochastic gradient descent with data augmentation) gracefully generalize and can withstand some erroneous supervision. Hence, these accuracy estimations may be overly pessimistic.

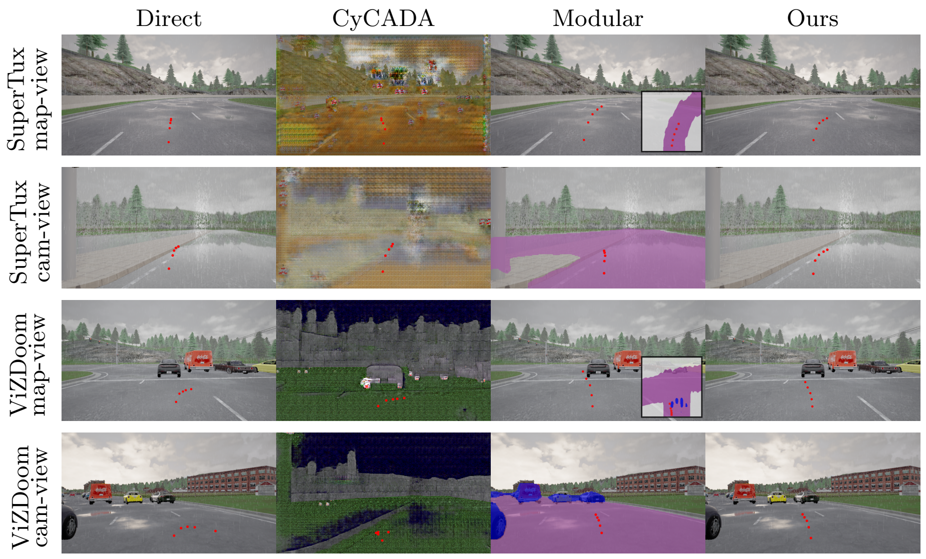


Fig. 4: We qualitatively examine how four different driving policies transfer to CARLA. Each policy is evaluated at the same state over four transfer methods, with predicted waypoints shown in red. Inferred modality is displayed for CyCADA and Modular. As shown, an inaccurate modality is used by a modular driving policy when transferring from SuperTuxKart via camera-view semantic segmentation. The median is misclassified as drivable road and the predicted waypoints direct the agent off of the road. (Best viewed on screen.)

4 Experiments and Results

We demonstrate our task distillation framework in two domain adaptation scenarios: 1) policy transfer of a navigation policy, and 2) general domain adaptation for semantic segmentation prediction.

4.1 Policy Transfer

Evaluating real-world navigational policies in a controlled and reproducible fashion can be tricky, requiring a physical vehicle and a reusable testbed environment. We sidestep these issues and instead transfer a navigation policy between three simulators of drastically different input fidelity, physical accuracy, and complexity: 1) SuperTuxKart, a simplistic open-source racing game, 2) ViZDoom [23], a professionally-developed maze-based shooting game, and 3) CARLA [10], a photorealistic driving simulator. As shown in Figure 3, the visual domain shift between these simulators is significant. We train our policies in the relatively low-fidelity ViZDoom and SuperTuxKart video games and transfer these agents to drive in the realistic CARLA driving simulator.

Method	Proxy task	Distance traveled (m)			Completion rate			
		avg.	min	max	100m	250m	500m	1000m
Direct	—	22.4 ±3.2	16.6	26.0	0.00	0.00	0.00	0.00
CyCADA [19]	—	24.0 ±1.3	22.4	26.0	0.00	0.00	0.00	0.00
CyCADA†	—	26.7 ±2.0	23.6	28.7	0.02	0.00	0.00	0.00
Modular§	cam-view	89.9 ±9.8	81.4	108.6	0.24	0.08	0.00	0.00
Modular [33]	cam-view	110.4 ±17.1	95.7	138.2	0.38	0.06	0.02	0.01
Ours	cam-view	164.6 ±14.9	147.5	191.6	0.59	0.18	0.03	0.00
Modular§	map-view	49.9 ±3.8	42.7	52.82	0.11	0.03	0.00	0.00
Modular	map-view	135.3 ±8.0	126.0	147.3	0.44	0.12	0.04	0.00
Ours	map-view	**260.5** ±15.2	**244.5**	**281.3**	**0.66**	**0.26**	**0.20**	**0.03**

Table 1: Adapting a SuperTuxKart racing agent to perform lane-following in CARLA. For each method, we evaluate 25 episodes using five fixed PID controller parameters. † denotes transferring raw low-level steering and throttle control to CARLA, as opposed to waypoints. § denotes training the driving policy using proxy task predictions in the source domain, as opposed to ground-truth labels.

We build our policies on the network architecture of Chen et al. [4], a ResNet-18 backbone that regresses to a trajectory plan of waypoints, which provide a domain-agnostic abstraction for control [33]. During deployment, a low-level PID controller converts waypoints to steering, throttle, and brake controls in CARLA.

We chose semantic segmentation in either camera-view [33] or map-view [2, 4, 55] (also known as bird's-eye view in the literature) as the proxy recognition representation. Semantic segmentation is a particularly useful proxy task since it is extremely prevalent in most datasets and domains. Moreover, it allows us to easily enforce relationships between domains simply by mapping source classes to target classes. In doing so, we can easily frame the desired behavior of the target CARLA vehicle in terms of the source agent we wish to transfer.

We build our policies on the network architecture of Chen et al. [4], wherein each policy outputs a trajectory plan using waypoints. This representation provides a domain-agnostic abstraction for control [33]. During deployment in the target domain, waypoints are fed into a low-level PID controller to obtain steering, throttle, and brake controls in CARLA.

Evaluation. For each transferred policy, we evaluate how far the agent travels until a driving infraction (i.e: collision) in an unseen test town in CARLA. We force all agents to travel at 20 km/h, as traveling speed greatly impacts the agent's performance. For waypoint-based agents, we hand-tune five PID controllers on a reference planner in a training town, and use these controllers to obtain low-level driving commands. We then evaluate all agents for 25 episodes under each controller, selecting the weather configuration and spawn points at random. Our experimental setup follows the official evaluation protocol of CARLA [10], except

Method	Proxy task	Distance traveled (m)			Completion rate			
		avg.	min	max	100m	250m	500m	1000m
Direct	—	17.4 ±2.4	13.6	21.0	0.02	0.00	0.00	0.00
CyCADA	—	24.4 ±5.1	20.7	33.9	0.02	0.00	0.00	0.00
Modular[§]	cam-view	148.6 ±40.3	92.6	211.2	0.42	0.20	0.01	0.00
Modular	cam-view	140.4 ±18.6	120.2	173.7	0.51	0.16	0.02	0.00
Ours	cam-view	166.9 ±33.6	125.2	223.8	0.62	0.17	0.06	0.00
Modular[§]	map-view	89.5 ±7.8	78.3	100.9	0.31	0.06	0.00	0.00
Modular	map-view	145.3 ±15.5	125.2	170.9	0.55	0.18	0.02	0.00
Ours	map-view	**277.3** ±56.6	**204.2**	**353.3**	**0.63**	**0.35**	**0.20**	**0.05**

Table 2: Results from transferring a ViZDoom maze-navigation agent to perform lane-following and vehicle-avoidance in CARLA.

we do not have a fixed goal, and thus do not use high-level commands. Instead, agents can chose to follow any route through the test town.

Baselines. We evaluate our method against direct transfer, image-to-image translation via CyCADA [19], and modular transfer [33]. Direct transfer ignores the issue of domain shift — we simply evaluate the source model in the target domain. Different baselines rely on varying proxy supervision during training, but the final model for each method maps raw input images to waypoints.

SuperTuxKart → CARLA. SuperTuxKart is a simple racing game, wherein players are expected to race on winding tracks — which can be quite challenging to traverse — by controlling the steering angle and throttle. Compared to the two-way traffic and major intersections found in CARLA environments, each SuperTuxKart map has a single non-overlapping one-way track. We train our source policy in SuperTuxKart using proximal policy optimization (PPO) [46] to maximize the distance traveled down the track. Our goal is to transfer this source policy to drive in CARLA. Following the setup of Müller et al. [33], we disable other traffic participants in CARLA, and focus solely navigating the road. We report our results in Table 1.

Ignoring the issue of visual domain shift and simply deploying our source policy in CARLA results in predictably poor results — the agent drives 100m without infraction only 2% of the time. Attempting to translate the target visual inputs to emulate the style of the source domain (as done in unsupervised domain adaptation approaches such as CyCADA) yields a poor and inconsistent inferred modality, as shown in Figure 4. Using this faulty input representation results in an agent with similarly low driving performance. As highlighted in Figure 3, these two visual domains are simply too far from each other.

By leveraging the compact semantic segmentation representation, the modular pipeline approach and our task distillation method both achieve an order of magnitude improvement in driving performance. For both sets of experiments, we align the "track" class in SuperTuxKart with the "road" class in CARLA to preserve the original source task's semantic relationship with the navigable track region (i.e: to stay on the track).

We observe that the map-view proxy task works better for both methods. As Wang et al. [55] show, this representation is more informative than the camera-view, but it is also harder to infer due to perspective distortion. Task distillation enables us to transfer a policy that is able to navigate twice as far as a modular approach using the same map-view proxy task. Moreover, the map-view results in a 66.2% increase in distance traveled over semantic camera-view under task distillation; however, it provides almost no improvement in a modular pipeline. Finally, unlike with a modular pipeline, our agent can exploit a stronger signal with being distracted by distortions, as they do not exist in the ground truth maps. See Figure 3 for visual examples.

Finally, we evaluate the performance of our transferred agent as the target-domain proxy-task dataset size changes (see Figure 5). By default, we use ten thousand labeled target images to both train the second stage of task distillation and to train the target recognition model in our modular pipelines. We find that even when training on only *half* of the target-domain samples, task distillation outperforms the modular approach.

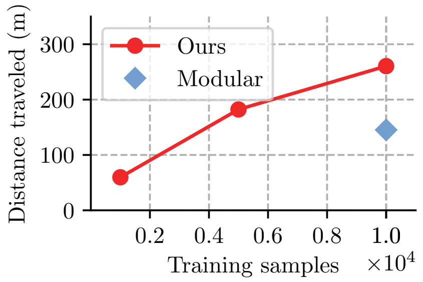


Fig. 5: Performance at different amounts of target-domain training data.

ViZDoom → CARLA. In ViZDoom, we train an agent using direct future prediction (DFP) [9] to navigate and explore complex maze environments while searching for health-kits and avoiding poison. Unlike the winding SuperTuxKart tracks, ViZDoom maze corridors resemble the road intersections of an urban driving environment. We aim to transfer this policy to a crowded street scenario in CARLA, wherein the agent must avoid other traffic participants while also navigating the road. To this end, we semantically align the two domains by mapping the "poison" class in ViZDoom with the "car" class in CARLA, in addition to aligning the "road" and "floor" classes. Hence, we frame the agent's tendency to avoid poison in the context of our target domain; i.e: the transferred policy avoids cars while staying on the road. We report our results in Table 2.

Direct transfer and CyCADA fail to yield a strong target policy and crash within a few meters. Despite facing a more difficult CARLA evaluation with traffic participants, both the modular approach and task distillation outperform their counterpart experiments in SuperTuxKart. We attribute these improvements to the inherent similarity between ViZDoom corridors and urban roads. Task distillation significantly exceeds the performance of a modular pipeline, especially when using the expressive, but difficult to infer, map-view proxy task.

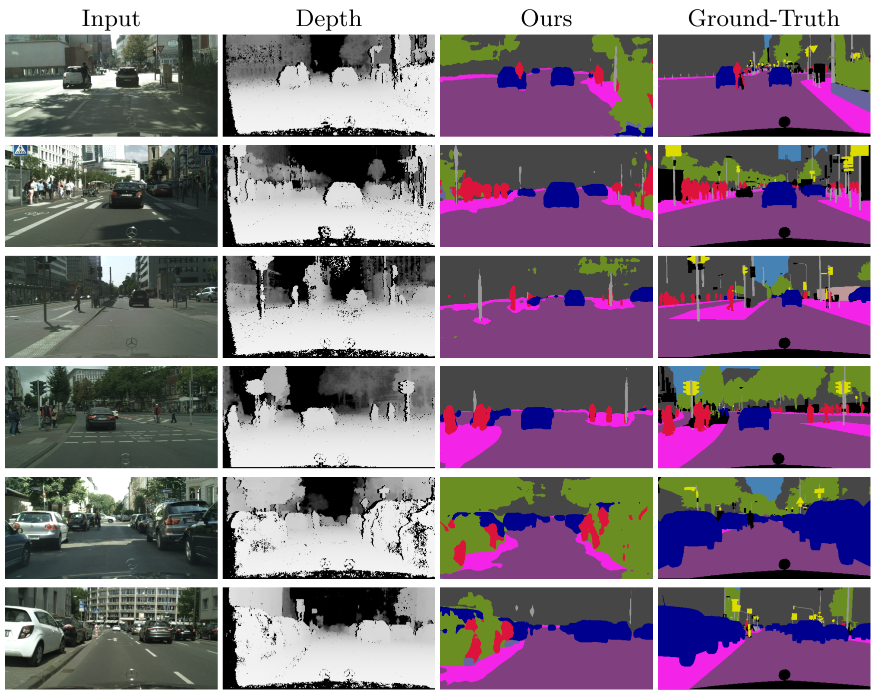


Fig. 6: Transferring the semantic segmentation task from the simulated SYNTHIA-SF dataset to the real-world Cityscapes dataset using task distillation with a depth estimation proxy task. We display representative samples near the reported mIoU in the top four rows. When depth estimation from stereo vision is too noisy, performance drops substantially, as shown in the bottom two rows. This highlights the importance of proxy task performance. Our final adapted model predicts semantic segmentation using only raw monocular images — we simply display the proxy depth labels for illustration.

Our policy transfer experiments concern a very specific domain adaptation application. However, task distillation is not limited to transferring sequential decision-making policies between domains. We can apply our framework to transfer general computer vision tasks from simulated to real-world datasets.

4.2 General Domain Adaptation

We transfer the semantic segmentation task between a variety of datasets: 1) SYNTHIA [17,39], a collection of synthetic scenes in a virtual city, 2) Cityscapes [6], an assortment of video sequences recorded on real-world urban streets, and 3) a hand-collected set of frames rendered in the CARLA driving simulator. All three datasets consist of raw monocular images with semantic and depth annotations.

	Source	Target	Method	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
(a)	SYNTHIA-SF	CARLA	Direct	9.1	30.9	0.1	0.0	13.7	14.9	57.3	15.5	4.5	22.8	64.6	21.2	48.0
	SYNTHIA-SF	CARLA	AT/DT¶	63.9	54.9	15.2	0.0	13.6	12.8	52.7	27.3	4.9	50.2	79.7	34.1	73.4
	SYNTHIA-SF	CARLA	AT/DT [37]	73.6	62.6	**26.9**	0.0	17.8	37.3	35.3	52.9	17.8	**63.0**	87.5	43.1	80.0
	SYNTHIA-SF	CARLA	Ours	**95.4**	**90.2**	10.4	0.0	**26.6**	**52.1**	**66.1**	**60.5**	**53.1**	57.3	**95.5**	**55.2**	**87.0**
(b)	SYNTHIA-SF	Cityscapes	Direct	0.7	1.6	0.0	0.0	13.4	6.0	**57.4**	21.6	**2.0**	42.7	19.7	15.0	41.3
	SYNTHIA-SF	Cityscapes	AT/DT¶	6.9	0.7	0.0	0.0	2.5	9.1	3.2	8.9	0.8	25.9	26.9	7.7	28.5
	SYNTHIA-SF	Cityscapes	AT/DT	85.8	29.4	1.2	0.0	3.7	14.6	1.9	8.9	0.4	**42.8**	**67.1**	23.2	64.0
	SYNTHIA-SF	Cityscapes	Ours	**86.8**	**46.3**	**8.1**	0.0	**34.9**	**19.2**	12.9	**52.4**	0.3	40.9	6.4	**28.0**	**64.3**
	SYNTHIA-SF	Cityscapes	Proxy*	84.9	37.5	8.2	0.0	26.6	23.0	16.8	43.1	19.3	43.3	19.7	29.3	64.4

Table 3: Transferring a semantic segmentation prediction model between simulated and real-world datasets. ¶ denotes using the AT/DT architecture for direct transfer. * denotes evaluating our proxy model on the target dataset.

We train a semantic segmentation prediction model on the SYNTHIA-SF dataset and adapt the model on both the Cityscapes and CARLA datasets.

In our task distillation setup, we use the low-level visual signal of depth estimation as a proxy task. Our experimental setup mimics that of Ramirez et al. [37], who use depth as an auxiliary supervisory signal to link domains in their AT/DT framework. However, we utilize a different semantic segmentation network architecture, building upon the simple DeepLabv3 model [5].

Evaluation. As these datasets may have incompatible semantic classes, we merge these classes as per Ramirez et al. [37]. We use standard evaluation metrics for semantic segmentation, and report the intersection-over-union per class, mean IoU, and global pixel-wise accuracy. Although we rely on proxy labels during training, just as in our policy transfer experiments, our final adapted models predict semantic segmentation from raw images in the target domain.

SYNTHIA-SF $\rightarrow$ CARLA. As shown in Table 3, the AT/DT direct baseline (denoted by ¶) outperforms our direct baseline. As their baseline uses no auxiliary tasks for aiding transfer, this seems to suggest that our model architecture is inferior. However, task distillation is able to easily bridge this difference without any architectural changes. Using the same additional depth supervision, our simple DeepLabv3 model yields significant improvement in mIoU over the complex AT/DT architecture. In particular, we observe increase in the IoU of several pervasive semantic classes: "roads", "sidewalk", "vegetation", and "traffic sign."

SYNTHIA-SF $\rightarrow$ Cityscapes. Since the Cityscapes dataset contains noisy depth estimations from stereo images, whereas the SYNTHIA-SF dataset provides depth labels directly from the rendering engine, there is a non-negligible domain gap in label space. Moreover, stereo matching failures produce gaps in the Cityscapes depth estimation. Hence, we must carefully align the proxy labels to ensure strong transfer. To this end, we apply various data augmentations to make the SYNTHIA-SF depth maps more similar to the noisy Cityscapes labels. During training, we simulate this noise by randomly sampling Cityscapes masks and applying stereo-matching failure regions to the perfect depth maps in the source datasets. Furthermore, to encourage invariance to slight scale changes, we add per-pixel multiplicative noise to each image.

Despite this proxy label mismatch between the two datasets, our adapted model achieves an absolute improvement of 4.8 mIoU over Ramirez et al. [37], yielding improved accuracy in the "sidewalk", "person", and "vehicle" semantic classes, as shown in Figure 6. However, we struggle with the "sky" class, likely due to noisy depth labels. This result is quite exciting, as we do not depend on any specialized real-world sensors. Using simple stereo images, we can transfer a semantic segmentation model trained in simulation for use in the real world.

Our adapted model performs well on scenes with high-quality depth estimation and struggles otherwise. As our method thrives when the domain gap between label spaces is small, and hence better depth estimates would likely further improve final performance. Moreover, our final model performs similarly to the proxy model in the target domain, thereby indicating that 1) the final distillation is successful, and 2) our final model's performance is constrained by the inability of our proxy model to generalize to the noisy proxy labels.

Limitations. Task distillation, like the modular approach, relies heavily on the proxy task being transferable. For example, adapting a model from SYNTHIA-RAND, in which the camera poses vary drastically, to Cityscapes results in an mIoU of 11.5. Transfer via the depth estimation proxy task fails due to the inconsistent appearance induced by changes in camera pose.

5 Conclusion

In this work, we propose a simple and effective framework for transferring knowledge from a source to target domain. Our method doesn't require any end-task labels in the target domain. Instead, we choose a proxy task with ample annotations in both simulation and the real world, e.g., depth or semantics, to tie both domains together. Task distillation outperforms competing alternatives for simulation-to-reality navigation policy transfer and domain adaptation for semantic segmentation.

Acknowledgments. This work has been supported in part by the National Science Foundation under grant IIS-1845485.

References

1. Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al.: Solving Rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113 (2019)
2. Bansal, M., Krizhevsky, A., Ogale, A.: ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In: RSS (2019)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
4. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: CoRL (2019)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
8. Doersch, C., Zisserman, A.: Sim2real transfer learning for 3d human pose estimation: motion to the rescue. In: NeurIPS (2019)
9. Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. In: ICLR (2017)
10. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: CoRL (2017)
11. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge: A retrospective. In: IJCV (2015)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. In: IJCV (2010)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. In: TPAMI (2009)
14. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. In: IJRR (2013)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Hernandez-Juarez, D., Schneider, L., Espinosa, A., Vazquez, D., Lopez, A.M., Franke, U., Pollefeys, M., Moure, J.C.: Slanted stixels: Representing san franciscos steepest streets. In: BMVC (2017)
18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
19. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
20. Huang, H., Huang, Q., Krähenbühl, P.: Domain transfer through deep activation matching. In: ECCV (2018)
21. Huh, M., Agrawal, P., Efros, A.A.: What makes ImageNet good for transfer learning? arXiv preprint arXiv:1608.08614 (2016)
22. James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., Levine, S., Hadsell, R., Bousmalis, K.: Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In: CVPR (2019)

23. Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jaśkowski, W.: ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In: CIG (2016)
24. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: An interactive 3D environment for visual AI. arXiv preprint arXiv:1712.05474 (2017)
25. Krähenbühl, P.: Free supervision from video games. In: CVPR (2018)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NeurIPS (2012)
27. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., Ferrari, V.: The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982 (2018)
28. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: ICLR (2017)
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
30. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
31. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
32. Mousavian, A., Toshev, A., Fišer, M., Košecká, J., Wahid, A., Davidson, J.: Visual representations for semantic target driven navigation. In: ICRA (2019)
33. Müller, M., Dosovitskiy, A., Ghanem, B., Koltun, V.: Driving policy transfer via modularity and abstraction. In: CoRL (2018)
34. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: CVPR (2018)
35. Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-real transfer of robotic control with dynamics randomization. In: ICRA (2018)
36. Pomerleau, D.A.: ALVINN: An autonomous land vehicle in a neural network. In: NeurIPS (1989)
37. Ramirez, P.Z., Tonioni, A., Salti, S., Stefano, L.D.: Learning across tasks and domains. In: ICCV (2019)
38. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: ICCV (2017)
39. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
40. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. In: IJCV (2015)
41. Rusu, A.A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., Hadsell, R.: Sim-to-real robot learning from pixels with progressive nets. In: CoRL (2017)
42. Sadeghi, F., Levine, S.: CAD2RL: Real single-image flight without a single real image. In: RSS (2017)
43. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied AI research. In: ICCV (2019)
44. Sax, A., Zhang, J.O., Emi, B., Zamir, A., Savarese, S., Guibas, L., Malik, J.: Learning to navigate using mid-level visual priors. In: CoRL (2020)
45. Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D scene structure from a single still image. In: TPAMI (2008)

46. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
47. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: ICCV (2019)
48. Sun, Y., Tzeng, E., Darrell, T., Efros, A.A.: Unsupervised domain adaptation through self-supervision. ICLR (2019)
49. Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., Urtasun, R.: MultiNet: Real-time joint semantic reasoning for autonomous driving. In: Intelligent Vehicles (2018)
50. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS (2017)
51. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR (2011)
52. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: CVPR (2018)
53. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
54. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: DADA: Depth-aware domain adaptation in semantic segmentation. In: ICCV (2019)
55. Wang, D., Devin, C., Cai, Q.Z., Krähenbühl, P., Darrell, T.: Monocular plan view networks for autonomous driving. In: ICRA (2019)
56. Wong, K., Wang, S., Ren, M., Liang, M., Urtasun, R.: Identifying unknown instances for autonomous driving. In: CoRL (2019)
57. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: CVPR (2018)
58. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
59. Zhou, B., Krähenbühl, P., Koltun, V.: Does computer vision matter for action? In: Science Robotics (2019)
60. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)