Severity-Aware Semantic Segmentation with Reinforced Wasserstein Training

Xiaofeng Liu[1,2*], Wenxuan Ji[1,3], Jane You[4], Georges El Fakhri[5], Jonghye Woo[5]

[1]Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, USA.
[2]Carnegie Mellon University, Pittsburgh, PA, USA.
[3]School of Artificial Intelligence, Nankai University, Tianjin, China.
[4]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.
[5]Massachusetts General Hospital, Harvard University, Boston, MA, USA.

*liuxiaofengcmu@gmail.com

Abstract

Semantic segmentation is a class of methods to classify each pixel in an image into semantic classes, which is critical for autonomous vehicles and surgery systems. Cross-entropy (CE) loss-based deep neural networks (DNN) achieved great success w.r.t. the accuracy-based metrics, e.g., mean Intersection-over Union. However, the CE loss has a limitation in that it ignores varying degrees of severity of pair-wise misclassified results. For instance, classifying a car into the road is much more terrible than recognizing it as a bus. To sidestep this, in this work, we propose to incorporate the severity-aware inter-class correlation into our Wasserstein training framework by configuring its ground distance matrix. In addition, our method can adaptively learn the ground metric in a high-fidelity simulator, following a reinforcement alternative optimization scheme. We evaluate our method using the CARLA simulator with the Deeplab backbone, demonstraing that our method significantly improves the survival time in the CARLA simulator. In addition, our method can be readily applied to existing DNN architectures and algorithms while yielding superior performance. We report results from experiments carried out with the CamVid and Cityscapes datasets.

1. Introduction

Semantic segmentation (SS) has been a critical vision-based task aiming to classify each pixel of an image into different semantic classes. For autonomous driving, automatic surgery system, robotics, and augmented reality and generation [47, 46, 42], it is an important way to precisely understand the visual scene. Benefited by the recent advances of deep learning [22, 35], a significant amount of effort has been devoted to this topic [48] in the past decades, leading to considerable progress on major open benchmark datasets [11]. Recently, the segmentation problem has been successfully tackled by pixel-wise classification based on the cross-entropy (CE) loss.

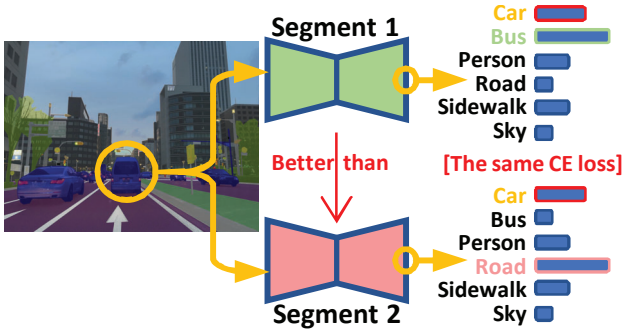

Figure 1. The cross-entropy loss has the same punishment for two softmax predictions (i.e., the same probability at i^* position), while these two segmentors can result in different severity consequences for real-world autonomous driving system.

However, the application of segmentors in many real-world tasks is still challenging, e.g., self-driving car, since they can have varying degrees of severity $w.r.t.$ different misclassification cases. For example, an accident of Tesla is caused by wrongly perceiving a white truck as the sky, arousing intense discussion of autonomous vehicle safety[1]. However, the result may be different if the system had just misclassified the truck as car or bus class.

As shown in Fig. 1, compared with the bottom segmentation prediction (Car→Road), the top one is more preferable (Car→Bus), while the cross-entropy loss does not discriminate these two softmax probability histograms. We note that with one-hot ground-truth label, the cross-entropy loss is only related to the prediction probability of the true class

[1]https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html

p_{i^*}, where i^* is the index of the true class. More formally, $\mathcal{L}_{CE} = -\log p_{i^*}$.

Actually, there are severity correlations of each label class, $e.g.$, severity(Car→Bus)>severity(Car→road) and severity(Person→Road)>severity(Sky→Road). When using the cross-entropy loss, the segmentation classes are considered independently [36], and the pair-wise inter-class correlations are not taken into account.

Our claim is also closely related to the importance-aware segmentation/classification [7, 10, 26]. These methods propose to define some class groups based on the pre-defined importance of each class. For example, the person and car are the most important classes, while the road and sidewalks are the less important classes, and the sky is the least important group. [7] simply assign a larger weight to the more important pre-defined group to calculate the weighted sum loss. Therefore, the system that misclassifies a pixel of a person into any other classes will get larger loss than misclassifying a pixel of the sky into any other classes. This is a nice property, but not sufficient for safe driving as it cannot discriminate the pair-wise severity of different predictions in misclassification cases as illustrated in Fig. 1.

To sidestep the aforementioned difficulties, in this work, we resort to the Wasserstein distance as an alternative for cross-entropy loss. The 1^{st} Wasserstein distance can be the optimal transport for transferring the probability masses from a source distribution to a target distribution [41]. For each pixel, we can calculate the Wasserstein distance between a softmax output histogram and the corresponding one-hot label, and we configure the ground metric as the severity misclassification. Therefore, it is possible to evaluate the softmax prediction of each pixel that is sensitive to the different misclassifications. The closed form solution of our Wasserstein loss with one-hot label follows the soft-attention setting as in [23] and could be fast computed using [26]. For semantic segmentation with unsupervised domain adaptation using constrained non-one-hot pseudo-label [48], we can also resort to the fast approximate solution of the Wasserstein distance.

In addition, instead of pre-defining the ground distance based on expert knowledge, we further propose to learn the optimal ground metric and a driving policy simultaneously in a high-fidelity autonomous driving simulator (e.g., CARLA) following an alternative optimization scheme. Our actor makes the decision based on the latent representation of the segmenter which is a partial observation of the front camera view. It can largely compress the state space for fast and stable training.

We summarize our contributions of this work as follows:

1) We point out a principled severity-aware semantic segmentation objective that has not been noticed by a trillion-dollar industry, i.e., autonomous driving. Instead of the importance-aware setting, it is necessary to discriminate the pair-wise misclassification severity (car→person≠car→bus), and the importance-aware methods can be a particular case by designing a specific ground metric. We believe our insights shed a light on the objective design of semantic segmentation tasks in the context of autonomous driving and surgery systems.

2) The pair-wise misclassification severity can be explored as a priori in our learned ground matrix in our Wasserstein training framework.

3) The ground metric can also be adaptively learned with a partially observable reinforcement learning (RL) framework based on a high-fidelity autonomous driving simulator following an alternative optimization scheme.

We demonstrate its effectiveness and generality on multiple challenging benchmarks with different backbone models and achieve promising performance on the high fidelity CARLA simulator.

2. Related Works

Semantic segmentation aims to describe the category, location, and shape [3]. With the development of deep neural networks [29, 20, 6], [37] propose to use the fully convolutional network (FCN) for the pixel-wise classification. The widely adopted cross-entropy loss in deep learning frameworks assign the same the loss for different error [21, 28, 32], which however does not consider the different severity results of different pair-wise mistakes.

Recently, [8, 26] propose that the different importance between classes (i.e., the importance-aware settings) should be taken into account. The categories in Cityscapes can be grouped based on their manually defined importance. The loss of each pixel in more important classes (e.g., in groups 3 and 4) will be given larger weights to compute the sum of loss in all pixels. Therefore, the misclassification of a pixel with a ground truth label in group 4 will result in a larger loss than misclassifying the sky to the other classes. However, its class-correlation is only defined in the ground truth perspective rather than prediction classes. Receiving the same loss by recognizing a car to bus or road is not sufficient for reliable autonomous driving. Essentially, they simply use a larger weight for the more important group's pixel when calculating the sum of loss in an image. Our more general severity-aware setting can explicitly discriminate the pair-wise mistake. In fact, the recently developed importance-aware segmentation methods [8, 26] are a special but inferior setting.

Besides, the grouping manipulation is only based on expert knowledge, which may differ from the way that a machine perceives the world [30, 27]. Our ground metric can be adaptively learned in an RL framework with an alternative training scheme.

Moreover, LRENT [48] is a method to overcome the unreliable pseudo label in self-training based domain adapta-

tion. That work proposes to smooth the one-hot label of each pixel. We further systematically investigate the possible fast solution with the conservative label.

Recently several advanced deep segmentation networks [9] and the pose-processing solutions have been developed [19]. Please note that these works are orthogonal with our framework and can be simply added following a plug-and-play fashion.

Wasserstein distance is used to measure the discrepancy of two distributions [26]. It has attracted much attention in the area of generative learning in particular [2]. [14] propose to use the Wasserstein distance for multi-label classification. Our previous works [25, 33] adopt the Wasserstein loss for ordinal classification (i.e., multi-level medical diagnosis) and the modulo classification (i.e., pose estimation). The ground matrix follows a specific ordinal/modulo constraint and can be solved with the fast exact solution. Recently, we further apply it as an alternative of importance-aware semantic segmentation [26]. Noticing that the tree structure used in [14, 26] follows the simple symmetric matrices, while $\mathbf{D}$ is asymmetric in our scenario as shown in Fig 2.

Based on the above fundamental methods, we propose to apply it to the severity-aware SS, and to ecode the severity of pair-wise misclassification with the ground matrix.

Reinforcement learning considers how the agent should take into account a specific environment state to maximize its cumulative reward [31]. The dynamic environment is usually stated with the Markov decision process. Recently, the advanced deep RL achieved human-level performance in many tasks, e.g., Atari Games [39] and Go.

End-to-end vision-based autonomous driving models [12] trained by RL usually have a high computational cost. To sidestep this issue, [38] propose to use variational inference to estimate policy parameters, while simultaneously uncovering a low-dimensional latent space of actors. Similarly, [15] analyze the utility of hierarchical representations for reuse in related tasks while learning latent space policies for RL.

We propose that the bottleneck of segmenter can be a natural representative lower-dimensional latent space which can efficiently shrink the state space and which requires fewer actor parameters. Besides, we incorporate RL in an alternative optimization framework to learn the optimal ground matrix in a simulator with a certain reward rule.

3. Methodology

In this section, we elaborate on our proposed approach, and target to render reliable segmentation results for autonomous driving by considering the different severity of pair-wise misclassification.

In the semantic segmentation task, we propose to learn a segmenter h_w, parameterized by w, with an autoencoder

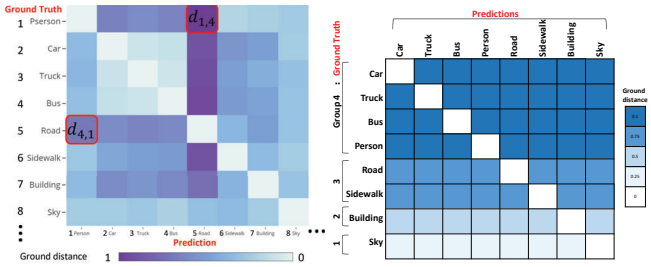

Figure 2. Left: a possible ground matrix for severity-aware segmentation. Right: the ground matrix as an alternative for the importance-aware setting [26].

structure. Let $\mathbf{s} = \{s_i\}_{i=1}^{N}$ be the prediction of a pixel in $h_w(\mathbf{X})$, $\mathbf{t} = \{t_j\}_{j=1}^{N}$ be the target label, and N be classes probability normalized by the softmax function. Let $i \in \{1, \cdots, N\}$ be the index of segmentation class. The learning is performed on the hypothesis space $\mathcal{H}$ of h_w. For the training example $\mathbf{X}$ and the corresponding target label $\mathbf{T} \in \mathbb{R}^{M_s \times M_s \times N}$, learning is achieved by minimizing $\min_{h_w \in \mathcal{H}} \mathcal{L}(h_w(\mathbf{X}), \mathbf{T})$. The loss function $\mathcal{L}(\cdot, \cdot)$ is used as an alternative for the performance measure. Typically, loss in SS is the sum of pixel-wise error.

Unfortunately, cross-entropy loss simply treats each class probability independently [14], ignoring the different severity of pair-wise misclassification.

Assuming the elements $\mathbf{D}_{i,j}$ indicate the pair-wise severity of misclassifying i-th class pixel into j-th class. In the classification setting, $\mathbf{s}$ and $\mathbf{t}$ are the histogram distributions. The closed form solution of the Wasserstein loss [26] can be formulated as

$$\mathcal{L}_{\mathbf{D}_{i,j}}(\mathbf{s}, \mathbf{t}) = \inf_{\mathbf{M}} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \mathbf{D}_{i,j} \mathbf{M}_{i,j}, \tag{1}$$

where $\mathbf{M}$ is the moving weights matrix, and its elements $\mathbf{M}_{i,j}$ is the to be moved masses from the i^{th} position in one histogram (e.g., softmax normalized output) to the j^{th} position in another histogram (e.g., target label). A valid moving weights matrix, i.e., $\mathbf{M}$ should subject to: $\mathbf{M}_{i,j} \geq 0$; $\sum_{j=0}^{N-1} \mathbf{M}_{i,j} \leq s_i$; $\sum_{i=0}^{N-1} \mathbf{M}_{i,j} \leq t_j$; $\sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \mathbf{M}_{i,j} = \min(\sum_{i=0}^{N-1} s_i, \sum_{j=0}^{N-1} t_j)$ [26].

The elements of ground distance matrix $\mathbf{D}_{i,j}$ for our application are illustrated in Fig. 2. For instance, classifying a car into the road ($d_{2,5}$) has a larger ground distance than a car into a bus ($d_{2,4}$).

Eq. 1 can be the optimal transportation distance if the two histograms with the identical probabilities sum, $\sum_{i=0}^{N-1} s_i = \sum_{j=0}^{N-1} t_j$, the ground metric $d_{i,j}$ should be symmetrical w.r.t. the main diagonal as $\mathbf{D}_{i,j}$. This is satisfied in [34, 25, 14]. However, this is not true for the severity-aware setting. For example, classifying a person into the road can be much severe than classifying the road into a person. Therefore, in Fig. 2 left, $d_{1,4}$ should have a

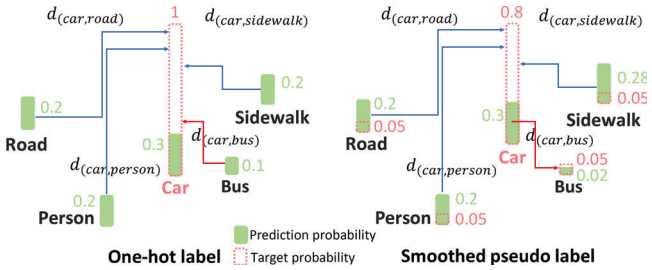

Figure 3. Left: There is only one transportation route in the one-hot setting [26]. Right: the transportation plan with conservative pseudo labels is more complicated, $e.g.$, car→bus.

larger value than $d_{4,1}$. Noticing that the importance-aware setting can be achieved by configuring the ground matrix as Fig. 2 right, which does not discriminate the different mistakes, $e.g.$, classifying the car into any other classes has the same punishment. The groups are also manually predefined, which however may not necessarily be appropriate for the practical driving system.

3.1. Wasserstein loss

With the one-hot label, the target histogram can be $\mathbf{t} = \delta_{j,j^*}$, and j^* indicates the segmentation category. δ_{j,j^*} is a delta function, and $\delta_{j,j^*} = 1$ for $j = j^*$ and $\delta_{j,j^*} = 0$ otherwise. Assuming $\sum_{j=0}^{N-1} t_j = \sum_{i=0}^{N-1} s_i$, and $\mathbf{t}$ with $t_{j^*} = 1$(or $\sum_{i=0}^{N-1} s_i$)[2], we have the only possible moving route as illustrated in Fig. 3 [26].

Therefore, the Wasserstein loss in Eq. 1 can be simplified to

$$\mathcal{L}_{\mathbf{D}^f_{i,j}}(\mathbf{s}, \mathbf{t}) = \sum_{i=0}^{N-1} s_i f(d_{i,j^*}). \tag{2}$$

[26] propose to extend $\mathbf{D}_{i,j}$ to $f(d_{i,j})$, where f can be a monotonic increasing mapping function [34].

The complexity of its closed form solution is $\mathcal{O}(N)$. Actually, our ground metric $f(d_{i,j^*})$ can be regard as weights of s_i, and follows a soft attention scheme [31]. The cross-entropy loss $-1\log s_{j^*}$ can be regarded as the hard prediction scheme [23], meaning that the other class's predictions are simply discarded, which results in a large information loss [31].

When we use the conservative target label, Eq. (2) does not apply. The exact solution has complexity higher than $\mathcal{O}(N^3)$. Therefore, a possible way is to resort to its approximation solution with complexity $\mathcal{O}(N^2)$ [26, 34].

[2]Noticing that with the rounding operation, the softmax normalization outputs cannot strictly be the sum of 1. However, setting t_{j^*} to 1 or $\sum_{i=0}^{N-1} s_i$) does not result in significant difference when we accurate to 8 decimal places.

3.2. Learn severity-aware ground matrix

Other than the pre-defined $\mathbf{D}$, this section proposes to adaptively adjust ground matrix in a simulator using the autonomous driving agent with a self-learning algorithm.

We design a novel alternative training framework to adaptively learn the ground matrix $\mathbf{D}$. It is possible to go beyond the expert knowledge following the recent advances in convolution neural networks. Its combination with RL with the ISO standard can further open the way to the end-to-end training without the need to design neither an evaluation metric (e.g., mIoU) nor $\mathbf{D}$, finally, achieving the "meta" learning.

The overall framework of the proposed system is illustrated in Fig. 4. We choose a high-reality simulator, CARLA [12], as our environment. The view of a monocular camera placed at the front of the car is rendered as $\mathbf{X}$. Segmenter takes $\mathbf{X}$ as input and predicts the segmentation image $\mathbf{S}$ which is compared with the target $\mathbf{T}$ with the Wasserstein loss.

An RL agent learns to interact with the environment following a partially observable Markov decision process (POMDP) [31, 24]. For every time step t, it takes a state s_t in a state space $\mathcal{S}$ as input and predict the action a_t from the action space $\mathcal{A}$, according to the RL policy $\pi(a_t|s_t)$ (i.e., the behavior of the agent) [31]. Then the action will result in the next environment state s_{t+1} in the dynamic system, and receive a reward $r_t(s_t, a_t) \in \mathcal{R} \subseteq \mathbb{R}$. The objective of RL is to find the optimal policy π^* to maximize the expectation of the weighted sum of rewards $R_t = \sum_{i\geq 0}^{T} \gamma^i r_{t+i}(s_t, a_t)$, where $\gamma \in [0, 1)$ indicates the discount parameter. It is used to balance the current and the long-term returns [18, 31].

Instead of using $\mathbf{X}$ as our state [12], we propose to utilize the latent representation of our segmenter. It can be either feature vector or feature maps according to the backbone. A recent work [12] takes 12 days for the training on CARLA with only 84×84 size raw image. As a partial observation, the latent representation compresses the state space drastically. Compared to the raw image, the segmentation map or its latent representation has sufficient information ($e.g.$, each object and precise location) to guide the driving, and is robust to appearance variation ($e.g.$, weather, lighting, etc.). Since a high proportion of pixels have the same label as their neighbors in $\mathbf{S}$, there is a large room to reduce its redundancy.

The input to the network is the concatenation of two latent representations from the two recent frames at this time step, as well as a vector of measurements ($e.g.$, sensor readings). They are inputted into two separate networks, i.e., the fully-convolutional network for feature maps, and a fully-connected network for the measurements. After these two branches are fed in, their processed results are concatenated and fed into the latter networks.

In the context of autonomous driving, we define the ac-

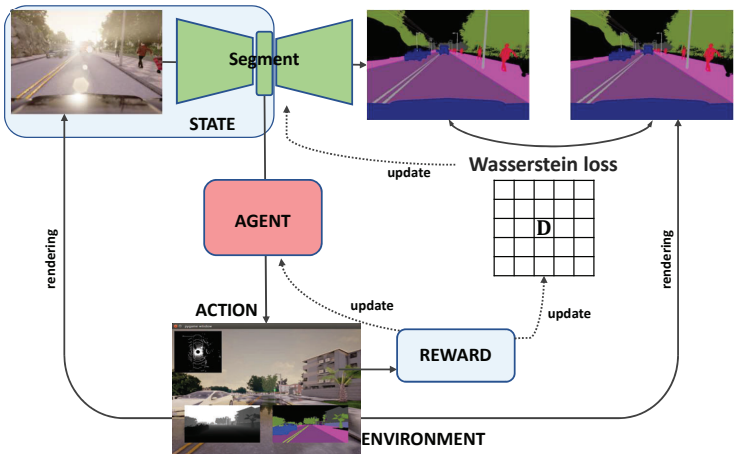

Figure 4. The reinforced alternative optimization framework to learn actor-critic agent and ground matrix simultaneously.

tion as a three-dimensional vector for steering $a_t^s \in [-1, 1]$, throttle $a_t^t \in [0, 1]$, and brake $a_t^b \in [0, 1]$. We define the reward $r_t = 1 - \alpha o_l - \beta o_r - \psi c$, where o_l and $o_r \in [0, 1]$ measure the degree of off-line and off-road, respectively, and $c \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ indicates that there is no/S0/S1/S2/S3 level crash, where S0, S1, S2, and S3 denotes the severity of negligible/minor, major, hazardous, and catastrophic, respectively, defined in [ISO26262] [16]. α, β, and ψ are a set of positive weights to balance the punishments, and we empirically set $\alpha = 1, \beta = 1$, and $\psi = 10$ in all of our experiments. The agent will receive the reward of 1 when the vehicle drives smoothly and keep in line and road. The driving will be terminated when there is a crash / completely (100%) off-line / 50% off-road / reaches 500 time steps.

Since our action space is continuous, we choose actor critic solution. Noticing that the value-based RL, *e.g.*, Q-Learning is not applicable here. The actor critic network is essentially a policy-based method, which is trained to find a parameterized policy $\pi_\theta(a_t|s_t)$ to maximize the expected long-term reward $J(\theta)$ [31]. According to the Theorem of Policy Gradient [43], the gradient of the parameters given the objective function can be:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t)(Q(s_t, a_t) - b(s_t))], \quad (3)$$

where $Q(s_t, a_t) = \mathbb{E}[R_t|s_t, a_t]$ is the state-action value function. The initial action a_t is given in order to compute the expected return when starting in the state s_t. We typically subtract a baseline function $b(s_t)$ to reduce the variance without changing the estimated gradient [45, 1]. A candidate for this baseline function is the state only value function $V(s_t) = \mathbb{E}[R_t|s_t]$, which is similar to $Q(s_t, a_t)$, except a_t is not given here. The advantage function can be expressed as $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ [18]. Eq.(4) then becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)]. \quad (4)$$

This can be a specific setting of the actor critic model in that $\pi_\theta(a_t|s_t)$ is the actor and the $A(s_t, a_t)$ is the critic. To reduce the number of required parameters, the parameterized temporal difference error $\delta_\omega = r_t + \gamma V_\omega(S_{s+1}) - V_\omega(S_s)$ is used to approximate the advantage function. We denote the parameter of actor and critic function with θ and ω, respectively. Noticing that most of the network parameters are shared in a mainstream neural network, followed by being separated into two branches for policy and value predictions, respectively. We further adapt the A3C to its off-policy version to stabilize and speed up our training [31].

After configuring our RL module, we propose the adaptive adjusting scheme of $\mathbf{D}$ for the training of the RL agent using an alternative optimization framework [48, 34].

Step A: Maintaining the elements in $\mathbf{D}$ and calculate the loss $\mathcal{L}_{\mathbf{D}_{i,j}}(\mathbf{s}, \mathbf{t})$ to update the networks of the actor-critic module via back-propagation.

Step B: Maintaining the networks and post-processing ground matrix $\mathbf{D}$ with ℓ_1 distances in the feature level w.r.t. the segmentation classes.

In step B, the normalized activation map of the last convolutional layer is used at each point as a vector as it does not have subsequent non-linear units. Thus, averaging the feature vectors in each position that corresponds to the pixel on image-level with the same class label is appropriate to calculate the center and re-compute the $\mathbf{D}_{i,j}$ with the ℓ_1 distances of the centers $\overline{d}_{i,j}$. Targeting on stabilizing the training, we calculate $\mathbf{D}_{i,j} = \frac{1}{1+\kappa}\left\{f(\overline{d}_{i,j}) + \kappa f(d_{i,j})\right\}$ for every iterations. We linearly change the hyper-parameter κ from 10 to 0 in the training stage.

4. Implementation details

We configure the structure of our A3C agent following [31, 44]. The information encoded in our measurement vector incorporates the present state's vechle speed, remaining distance, collision damages, and the present high-level or-

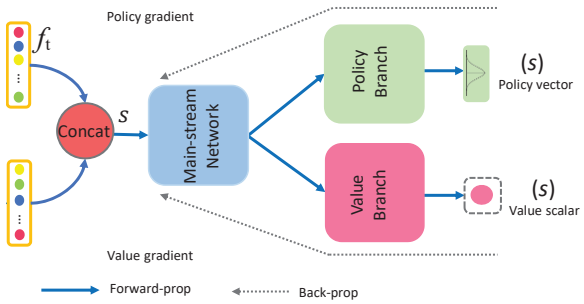

Figure 5. The feed-forward neural network architecture of the advantage actor-critic. The two input features are the processed feature map and the vector of measurements, respectively.

der given by the topological planner [12]. All of these vectors are in one-hot encoding. Our A3C is shown in Figure 8. It is trained using ten actor-thread parallelly. Totally 10,000,000 dynamic environment steps are implemented [12]. Besides, we also apply the 20-step roll-outs as the implementation of [17].

According to the CARLA simulator [12], the measurements used in our state are defined as the related information of the player's state and the simulator environment, e.g., position if the player, player speed, collision, opposite lane intersection, sidewalk intersection, the current in-game time, player acceleration, player orientation, sensor readings, non-client-controlled agents information, traffic lights information, and the speed limit signs information.

We adopt two fully connected (FC) layers (64, 64) to process the vector of measurements. We apply two convolutional layers with $3\times3\times32$ and $3\times3\times16$ kernels, followed by two fully connected layers (1024, 512). Since the latent feature map of different segmentation backbone has a different size, the trained network of this part cannot be shared among different backbones. As shown in Fig. 8, our actor-critic uses two FC layers (256, 128) alongside cascading branches which use the two FC layers (64, 16). The number of the output unit is set as 3, which indicates the steering, throttle, and brake.

The initial learning rate is set to 0.0007 and the entropy regularization to 0.01. Besides, the learning rate is gradually drop to 0 in the end. The evaluation details using the third-party reinforcement framework on CARLA including experiment settings, network structures, and hyperparameter settings are based on [13][3].

5. Experiments

Our framework is evaluated on the CARLA simulator [12] and two typical autonomous driving benchmarks (*i.e.*, Cityscapes [11] and CamVid [4]). To demonstrate the effectiveness of the learned ground matrix, we give a series of ex-

[3] https://gitlab.com/grant.fennessy/rl-carla

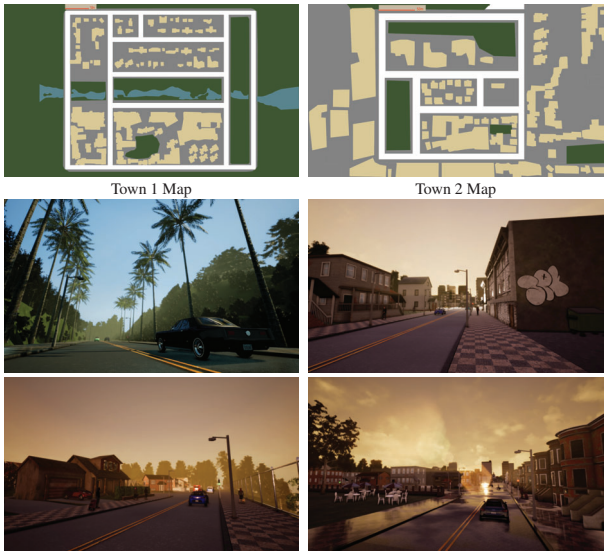

Figure 6. The two owns in CARLA simulator [5], where the left is the views and a map of CARLA Town 1 used for training. The right is the views and a map of CARLA Town 2 used for new town testing.

periments with different backbones. All of the experiments are pre-trained with the CE loss as their vanilla version.

We use the conventional intersection-over-union (IoU) $\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}}$ for the more important group that defined in [26, 7], where TP, FP, and FN denote the numbers of true positive, false positive, and false negative pixels, respectively. Moreover, the mean IoU is the average of IoU among all classes. However, these metrics cannot discriminate the different severity of pair-wise misclassifications.

We further adapt the third-party evaluations used in CARLA [13, 12]: **Drive%**: The drive% measures the number of steps that took place during the evaluation divided by 720,000. A value of 100% implies that the agent never had an early termination (due to stagnation, off-road, or collision), while a lower value implies some degree of failures. **Km**: Total kilometers driven across all steps in the evaluation. This is ultimately a function of mean speed and drive%. **Km/Hr**: Mean speed taken across all steps in the evaluation. Target speed is the pre-set maximum speed in CARLA 25km/hr. **Km/OOL**: How many kilometers are driven on average between each out of lane (OOL) infraction. An OOL infraction occurs any time the vehicle exits the lane in any way. A 2-second timer (100 steps) is kicked off after the infraction is detected, during which time no additional infractions can occur. Once the timer completes, a new infraction occurs if the vehicle is still in any way out of the lane. Wrapping up the out of lane infractions into these events helps to filter out instances where the vehicle just barely nudges out of the lane several times in rapid succession. Ideally, this value is infinite if no OOL instances

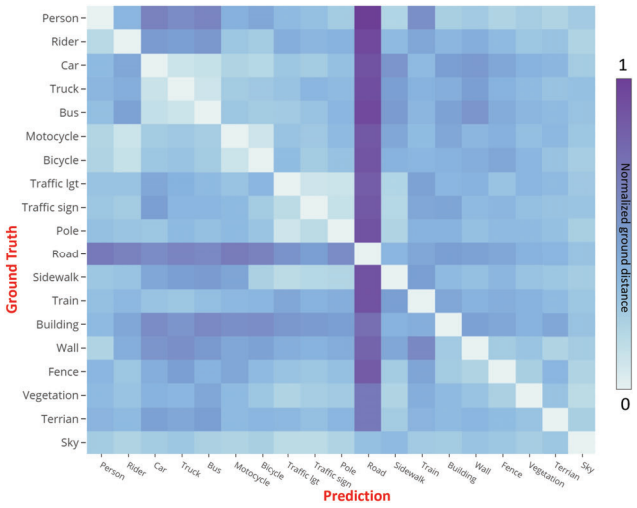

Figure 7. Illustration of adaptively normalized ground matrix learned in the CARLA simulator with the ENet backbone.

occur. **Km/Collision**: How many kilometers are driven on average between each collision with an object in the environment. Ideally, this value is infinite if no collisions occur.

5.1. Severity-aware SS with the learned ground matrix

As discussed in our introduction, the importance-aware setting does not consider the different severity $w.r.t.$ the predictions. Instead of pre-defining a severity-aware ground matrix with human knowledge, we propose to learn it in the CARLA simulator (as shown in Figure 6)[4] and show our result with the ENet backbone. We train our actor critic with ten threads parallelly as in [12]. The joint learning of our actor-critic module and the ground matrix only takes 10.5 hours, which is much faster than using the images as the state. We note that in [12], it takes 12 days to train an RL framework. The time cost will be intractable when we incorporate a ground matrix simultaneously. Besides, the Wasserstein training outperforms the IAL and vanilla ENet baseline w.r.t. the episode rewards by a large margin consistently. Higher episode rewards are expected for good learning algorithms.

CARLA offers a fine-grained evaluation of driving polices which characterizes the approaches by the average distance between different collisions and more than 30% off-line or off-road as follows:

Off-line: if over 30% of the car's footprint is over wrong-way lanes.

Off-road: if over 30% of the car's footprint is over the sidewalk.

Collision-static/car/person: if a car makes contact with a static object, another car and pedestrian respectively.

[4] https://carla.org

Task	Training condition		New town		New weather	
	wo/	w/	wo/	w/	wo/	w/
collision-person	12.61	**30.43**	2.53	**7.82**	9.24	**28.25**
collision-car	0.84	**4.59**	0.40	**2.79**	0.75	**4.33**
collision-static	0.45	**1.36**	0.26	**1.02**	0.28	**1.29**
off-line	0.18	**0.85**	0.21	**0.78**	0.14	**0.81**
off-road	0.76	**1.47**	0.43	**1.22**	0.71	**1.35**

Table 1. Average distance (in kilometers) traveled between two infractions of using the ENet trained only with the CE loss (wo/) or fine-tuned with the Wasserstein loss (w/) in our RL framework. The higher values indicate superior performance.

Method	Drive%	Km	Km/Hr	Km/Off-line	Km/Collision
Deeplab wo/	82.2	31.9	9.3	0.04	12.4
Deeplab w/ IAL	85.8	35.2	12.4	0.08	15.7
Deeplab w/ A-$\mathcal{L}_{d_{i,j}}$	**91.6**	**47.5**	**20.4**	**0.14**	**20.7**

Table 2. Results of different training methods using the Deeplab backbone [13] evaluation on the CARLA simulator. The higher values indicate superior performance.

Of note, the duration of each violation is limited to 2 seconds. The results are reported in Table 1. Rather than testing on the same town environment, we also test at a new town or new weather condition following the standard evaluation of CARLA. As expected, our method can largely improve these metrics and lead to a more safe driving system. By emphasizing the severity of the misclassification of a person, the average distance between two collisions with a person almost doubled in all of the testing cases.

Other than using our RL framework to make the driving decision, we also evaluate our segmented results using an independent autonomous driving system. [13] propose to process the front view image in CARLA with Deeplab [9] to get a segmentation and then combine it with the depth camera and vehicle stats as state. We replace its vanilla Deeplab module with a fine-tuned one using the Wasserstein loss or IAL. Following the experiment setting and evaluation metrics, we give the comparison in Table 2. The "A-" indecates the adaptive ground matrix adjusting and the learned matrix is shown in 7. The improvements over Deeplab and IAL trained Deeplab indicate that our segmenter can offer more reliable and safe segmentation results for the driving system.

5.2. Importance-aware SS with learned D

We can also apply our adaptively learned ground matrix to the importance-aware SS task. Following the standard IAL testing protocol [8, 7, 26], we apply the SegNet [3] and ENet [40] as our backbones.

For the Cityscapes dataset, Table 3 shows that the segmentation result of pixel in the categories in group 4 has higher IoU when considering the importance of each class. For the CamVid dataset, the results are reported in Table

	Group 4							mIoU
	Person	Rider	Car	Truck	Bus	Motor	Bike	
SegNet	62.8	42.8	89.3	38.1	43.1	35.8	51.9	57.0
+IAL	84.1	46.0	91.1	75.9	65.0	22.2	**65.3**	65.7
A-$\mathcal{L}_{d_{i,j}}$	85.4	47.4	90.3	76.9	69.3	41.5	62.4	65.9
ENet	65.5	38.4	90.6	36.9	50.5	38.8	55.4	58.3
+IAL	87.7	41.3	92.4	**73.5**	76.2	24.1	69.7	67.5
+A-$\mathcal{L}_{d_{i,j}}$	90.2	47.0	93.1	72.5	73.1	44.2	72.2	68.2

Table 3. The comparison of different loss functions on group 4 of Cityscapes dataset using the SegNet or ENet backbone.

	Group 3			Group 4			mIoU
	Road	Sidewalk	Sign	Car	Pedestrian	Bike	
FCN	98.1	89.5	25.1	84.5	64.6	38.6	69.6
+IAL	96.3	91.8	21.5	82.2	69.5	57.6	71.2
+A-$\mathcal{L}_{d_{i,j}}$	97.3	92.4	28.6	86.4	70.8	60.5	71.5

Table 4. The comparison of different loss functions on group 3/4 of CamVid dataset using the FCN backbone.

4. The Wasserstein training with the learned ground matrix can achieve comparable performance for groups 3 and 4 as the IAL methods. This indicates that the classes in groups 3 and 4 can play an important role for safe driving.

For the unsupervised domain adaptation with constrained self-training (LRENT) [48], we also use the approximate solution of the Wasserstein distance. The results GTA5→Cityscapes adaptation are given in Table 5. Our learned ground matrix can be applied to many real-world tasks.

Noticing that since we do not manage to achieve better performance in the IAL setting with the learned ground matrix, and the evaluation metrics used in IAL cannot demonstrate the superiority of our severity-aware setting, we give additional confusion statistics in Figure 8.

We can see that the prediction probability of SegNet+Wasserstein training more concentrates on car/truck/bus. Although the improvement of correctly classifying a car as a car is about 1% to 3% over IAL or SegNet as shown in Table 8, IAL/SegNet have more severe misclassifications, e.g., car→person and rider/motor/bike/sky. Noticing that since our correct classification probabilities in the other class are usually more significant and promising than the car, we just pick one that has the similar correct probability class and show how different they make mistakes. Even though they have the similar probability to be wrong, their consequences will have different severity.

6. Conclusions

We have introduced the severity-aware semantic segmentation setting, which is ignored by the previous works. The ground metric of our Wasserstein inspired loss indicates the pair-wise severity of misclassification and is learned by alternative optimization with an RL framework.

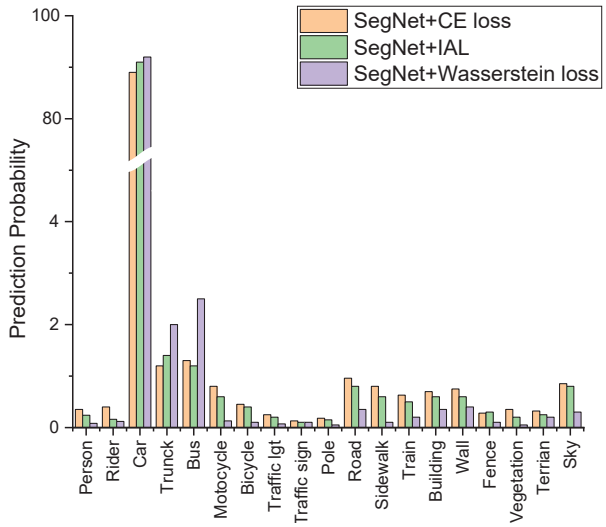

Figure 8. The confusion statistics of classifying a car on the testing set of Cityscapes dataset with the SegNet backbone.

	Group 4							mIoU
	Person	Rider	Car	Truck	Bus	Motor	Bike	
LRENT	61.7	27.4	83.5	27.3	37.8	30.9	41.1	46.5
A-$\mathcal{L}_{d_{i,j}}$	63.9	33.5	88.1	35.8	44.9	40.3	48.0	47.1

Table 5. The comparison of different loss functions on group 4 of GTA5→Cityscapes unsupervised domain adaptation using the DeeplabV2 backbone.

The importance-aware problem can be a special case of our framework. It has a simple exact fast solution in one-hot case and the fast approximate solution can be used for the conservative label in self-learning based unsupervised domain adaptation. We improve the autonomous driving metrics in the CARLA simulator significantly. Nevertheless, it is designed for semantic segmentation, we are possible to apply it to other problems with multiclass classification labels that have different severity of misclassification. In our future work, we plan to further consider the distance of the misclassified pixel to the observer. For example, a person misclassified as the road is much worse if the person is close to the driving vehicle than if it is still far away. A possible solution can be configuring via a depth estimation module with a camera/lidar and assigning a larger weight for the segmentation pixel near the observer.

7. Acknowledgement

The funding support from National Institute of Health (NIH), National Institute of Neurological Disorders and Stroke (NINDS) (NS061841, NS095986), Youth Innovation Promotion Association, CAS (2017264), Innovative Foundation of CIOMP, CAS (Y586320150) and Hong Kong Government General Research Fund GRF (Ref. No.152202/14E) are greatly appreciated.

References

[1] Alex M Andrew. Reinforcement learning: An introduction by richard s. sutton and andrew g. barto, adaptive computation and machine learning series, mit press (bradford book), cambridge, mass., 1998, xviii+ 322 pp, isbn 0-262-19398-1,(hardback,£ 31.95).-. Robotica, 17(2):229–235, 1999. 5

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017. 3

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. TPAMI, 39(12):2481–2495, 2017. 2, 7

[4] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2):88–97, 2009. 6

[5] Carlos A Cabrelli and Ursula M Molter. The kantorovich metric for probability measures on the circle. Journal of Computational and Applied Mathematics, 57(3):345–361, 1995. 6

[6] Tong Che, Xiaofeng Liu, Site Li, Yubin Ge, Ruixiang Zhang, Caiming Xiong, and Yoshua Bengio. Deep verifier networks: Verification of deep discriminative models with deep generative models. In ArXiv, 2019. 2

[7] Bike Chen, Chen Gong, and Jian Yang. Importance-aware semantic segmentation for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 20(1):137–148, 2018. 2, 6, 7

[8] Bi-ke Chen, Chen Gong, and Jian Yang. Importance-aware semantic segmentation for autonomous driving system. In IJCAI, pages 1504–1510, 2017. 2, 7

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI, 40(4):834–848, 2017. 3, 7

[10] Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. IEEE Transactions on Image Processing, 28(1):265–278, 2018. 2

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016. 1, 6

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938, 2017. 3, 4, 6, 7

[13] Grant Fennessy. Autonomous Vehicle End-to-End Reinforcement Learning Model and the Effects of Image Segmentation on Model Quality. PhD thesis, Vanderbilt University, 2019. 6, 7

[14] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In Advances in Neural Information Processing Systems, pages 2053–2061, 2015. 3

[15] Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, and Sergey Levine. Latent space policies for hierarchical reinforcement learning. arXiv:1804.02808, 2018. 3

[16] ISO26262 ISO. 26262: Road vehicles-functional safety. International Standard ISO/FDIS, 2011. 5

[17] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. arXiv preprint arXiv:1611.05397, 2016. 6

[18] Yuxi Li. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274, 2017. 4, 5

[19] Fayao Liu, Guosheng Lin, and Chunhua Shen. Crf learning with cnn features for image segmentation. Pattern Recognition, 48(10):2983–2992, 2015. 3

[20] Xiaofeng Liu. Research on the technology of deep learning based face image recognition. In Thesis, 2019. 2

[21] Xiaofeng Liu, Fangfang Fan, Lingsheng Kong, Zhihui Diao, Wanqing Xie, Jun Lu, and Jane You. Unimodal regularized neuron stick-breaking for ordinal classification. Neurocomputing, 2020. 2

[22] Xiaofeng Liu, Yubin Ge, Chao Yang, and Ping Jia. Adaptive metric learning with deep neural networks for video-based facial expression recognition. Journal of Electronic Imaging, 27(1):013022, 2018. 1

[23] Xiaofeng Liu, Zhenhua Guo, Site Li, Jane You, and Kumar B.V.K. Dependency-aware attention control for unconstrained face recognition with image sets. In ICCV, 2019. 2, 4

[24] Xiaofeng Liu, Zhenhua Guo, Jane You, and BVK Vijaya Kumar. Dependency-aware attention control for image set-based face recognition. IEEE Transactions on Information Forensics and Security, 15:1501–1512, 2019. 4

[25] Xiaofeng Liu, Xu Han, Yukai Qiao, Yi Ge, Site Li, and Jun Lu. Unimodal-uniform constrained wasserstein training for medical diagnosis. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019. 3

[26] Xiaofeng Liu, Yuzhuo Han, Song Bai, Yi Ge, Tianxing Wang, Xu Han, Site Li, Jane You, and Jun Lu. Importance-aware semantic segmentation in self-driving with discrete wasserstein training. In Proceedings of the AAAI, 2020. 2, 3, 4, 6, 7

[27] Xiaofeng Liu, Lingsheng Kong, Zhihui Diao, and Ping Jia. Line-scan system for continuous hand authentication. Optical Engineering, 56(3):033106, 2017. 2

[28] Xiaofeng Liu, BVK Vijaya Kumar, Ping Jia, and Jane You. Hard negative generation for identity-disentangled facial expression recognition. Pattern Recognition, 88:1–12, 2019. 2

[29] Xiaofeng Liu, Site Li, Lingsheng Kong, Wanqing Xie, Ping Jia, Jane You, and BVK Kumar. Feature-level frankenstein: Eliminating variations for discriminative recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 637–646, 2019. 2

[30] Xiaofeng Liu, Zhaofeng Li, Lingsheng Kong, Zhihui Diao, Junliang Yan, Yang Zou, Chao Yang, Ping Jia, and Jane You.

A joint optimization framework of low-dimensional projection and collaborative representation for discriminative classification. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 1493–1498. 2

[31] Xiaofeng Liu, BVK Vijaya Kumar, Chao Yang, Qingming Tang, and Jane You. Dependency-aware attention control for unconstrained face recognition with image sets. In Proceedings of the European Conference on Computer Vision (ECCV), pages 548–565, 2018. 3, 4, 5

[32] Xiaofeng Liu, BVK Vijaya Kumar, Jane You, and Ping Jia. Adaptive deep metric learning for identity-aware facial expression recognition. In CVPR Workshops, pages 20–29, 2017. 2

[33] Xiaofeng Liu, Yang Zou, Tong Che, Peng Ding, Ping Jia, Jane You, and B.V.K. Vijaya Kumar. Conservative wasserstein training for pose estimation. In The IEEE International Conference on Computer Vision (ICCV), October 2019. 3

[34] Xiaofeng Liu, Yang Zou, Tong Che, Jane You, and Kumar B.V.K. Conservative wasserstein training for pose estimation. In ICCV, 2019. 3, 4, 5

[35] Xiaofeng Liu, Yang Zou, Lingsheng Kong, Zhihui Diao, Junliang Yan, Jun Wang, Site Li, Ping Jia, and Jane You. Data augmentation via latent space interpolation for image classification. In 24th International Conference on Pattern Recognition (ICPR), pages 728–733, 2018. 1

[36] Xiaofeng Liu, Yang Zou, Yuhang Song, Chao Yang, Jane You, and BV K Vijaya Kumar. Ordinal regression with neuron stick-breaking for medical diagnosis. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018. 2

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431–3440, 2015. 2

[38] Kevin Sebastian Luck, Joni Pajarinen, Erik Berger, Ville Kyrki, and Heni Ben Amor. Sparse latent space policy search. In AAAI, 2016. 3

[39] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529, 2015. 3

[40] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. ICLR, 2017. 7

[41] Ludger Rüschendorf. The wasserstein distance and approximation theorems. Probability Theory and Related Fields, 70(1):117–129, 1985. 2

[42] Yuhang Song, Chao Yang, Zhe Lin, Hao Li, Qin Huang, and C-C Jay Kuo. Image inpainting using multi-scale feature image translation. 1

[43] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In NIPS, pages 1057–1063, 2000. 5

[44] Mnih Volodymyr, Kavukcuoglu Koray, Silver David, A Rusu Andrei, and Veness Joel. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015. 5

[45] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In Reinforcement Learning, pages 5–32. Springer, 1992. 5

[46] Chao Yang, Xiaofeng Liu, Qingming Tang, and C-C Jay Kuo. Towards disentangled representations for human retargeting by multi-view learning. arXiv preprint arXiv:1912.06265, 2019. 1

[47] Chao Yang, Yuhang Song, Xiaofeng Liu, Qingming Tang, and C-C Jay Kuo. Image inpainting using block-wise procedural training with annealed adversarial counterpart. arXiv preprint arXiv:1803.08943, 2018. 1

[48] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. ICCV, 2019. 1, 2, 5, 8