Wasserstein Loss with Alternative Reinforcement Learning for Severity-Aware Semantic Segmentation

Xiaofeng Liu, Yimeng Zhang, Xiongchang Liu, Song Bai, Site Li, and Jane You

***Abstract*—Semantic segmentation is important for many real-world systems, e.g., autonomous vehicles, which predict the class of each pixel. Recently, deep networks achieved significant progress w.r.t. the mean Intersection-over Union (mIoU) with the cross-entropy loss. However, the cross entropy loss can essentially ignore the difference of severity for an autonomous car with different wrong prediction mistakes. For example, predicting the car to the road is much more servery than recognize it as the bus. Targeting for this difficulty, we develop a Wasserstein training framework to explore the inter-class correlation by defining its ground metric as misclassification severity. The ground metric of Wasserstein distance can be pre-defined following the experience on a specific task. From the optimization perspective, we further propose to set the ground metric as an increasing function of the pre-defined ground metric. Furthermore, an adaptively learning scheme of the ground matrix is proposed to utilize the high-fidelity CARLA simulator. Specifically, we follow a reinforcement alternative learning scheme. The experiments on both CamVid and Cityscapes datasets evidenced the effectiveness of our Wasserstein loss. The SegNet, ENet, FCN and Deeplab networks can be adapted following a plug in manner. We achieve significant improves on the predefined important classes, and much longer continuous play time in our simulator.**

***Index Terms*—Semantic Segmentation, Autonomous Driving, Wasserstein Training, Actor-Critic.**

I. Introduction

Semantic segmentation (SS) has been an important computer vision task, which aiming to densely predict the discrete class labels of the pixel of image [63], [65]. For an autonomous driving, robotics, augmented reality and automatic surgery system, it is an important way to precisely understand the scene. Recently, many work have been done in this area [70], [1], and leading to considerable progress on major open benchmark datasets [12] with the advances of deep learning technology. In the deep learning era [28], [31], [19], [19], [30], [38], segmentation is essentially making the pixel-wise classification based on cross-entropy (CE) loss.

Unfortunately, the aforementioned models can encounter challenges in many practical applications such as autonomous driving, where one has different severity w.r.t. different misclassification cases. For example, an accident of Tesla is caused by a wrong recognition of a white truck as sky, arousing intense discussion of autonomous vehicle safety*. However, the result may have been different had just recognized the truck as car/bus. Similarly, Uber's car misclassified a person and finally resulted in a pedestrian being killed†.

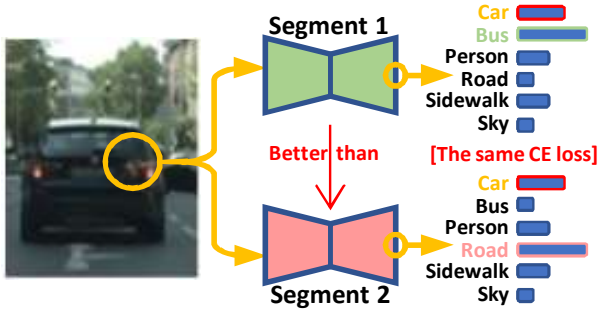

Fig. 1: The limitation of CE loss for real-world autonomous driving system. The true class of these pixels are car i^*. We show two softmax output of the segmenters which have the same probability at i^* position. They will be assigned with the same cross-entropy loss, while the first distribution can be more preferable than the second one. These two results can lead to different severity consequences.

As illustrated in Fig. 1, compared with the bottom segmentation prediction (Car→Road), the top one is more preferable (Car→Bus), while the CE loss does not discriminate these two softmax probability histograms. We note that with one-hot ground-truth label, the CE loss is only related to the prediction probability of the true class 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 classes $e.g.$, severity(Car→Bus)>severity(Car→road) and severity(Person→Road)>severity(Sky→Road). When using the cross-entropy objective, the classes are independent to each other [43], and the inter-class relationships are not been considered.

Our claim is also closely related to the importance-aware classification/segmentation [7], [64]. These methods were proposed to define some class groups based on the pre-defined importance of each class. For example, the car, truck, bus are in the most important group, road and sidewalks are

Xiaofeng Liu is with the Harvard University, Cambridge, MA, 02138 USA (Corresponding author: xliu11@bidmc.harvard.edu).

Yimeng Zhang is with the Columbia University and Harvard University, Cambridge, MA, 02138 USA.

Xiongchang Liu is with the China University of Mining and Technology and Harvard University, Cambridge, MA, 02138 USA.

S. Bai is with the Department of Statistics, the University of California Berkeley, Berkeley, CA, 94720 USA.

Site Li is with the Carnegie Mellon University, Pittsburgh, PA, 15232 USA.

J. You is with the Dept.of Computing, The Hong Kong Polytechnic University, Hong Kong.

Manuscript accepted Sep 27, 2019

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

†https://nypost.com/2019/11/07/

in the less important group, and the sky is in the least important group. Then, a larger weight will be multiplied to the more important group to calculate the loss. Therefore, misclassifying a car as any other classes will receive larger punishment than misclassifying the sky as any other classes. This is a nice property, but not sufficient for safe driving as it cannot discriminate the severity of different prediction in misclassification cases. $e.g.$, Fig. 1.

Targeting the aforementioned difficulties, we choose the Wasserstein distance as the alternative optimization objective. The first order Wasserstein distance can be regard as the optimal transport cost of moving the mass in one distribution to match the target distribution [56]. In this paper, we propose to calculate the Wasserstein distance between the softmax prediction of segmentor and its target label. We note that both of them are the normalized histograms. By setting the ground matrix as the misclassification severity, we are able to measure the prediction that sensitive to the pair-wise misclassifications.

The ground matrix of Wasserstein distance can be pre-defined with the experience to explore the pair-wise class correlation, $e.g.$, the divergence of car and road is larger than car and bus. From the optimization perspective, we also set the ground metric to its increasing function. For semantic segmentation with unsupervised domain adaptation using constrained non-one-hot pseudo-label, we can also resort to the fast approximate solution of Wasserstein distance.

Instead of pre-defining the ground metric based on expert knowledge, we further propose to learn the optimal ground metric and a driving policy simultaneously in the CARLA simulator with an alternative optimization scheme. Our actor makes decision based on the latent representation of segmenter which is a partial observation of the front camera view. It can largely compress the state space for fast and stable training.

This paper is an extension of our preliminary segmentation work [33], [34]. In summary, the contributions of this paper are summarized as

• We propose to render reliable segmentation results for autonomous driving by considering the different severity of misclassification. The inter-class severity is explicitly incorporated in the ground metric of our Wasserstein training framework. The importance-aware methods can be a particular case by designing a specific ground metric.

• The ground matrix can also be adaptively learned with an partially observable reinforcement learning framework based on the autonomous driving simulator with the alternative optimization.

• For both the one-hot and non-one-hot target label in self-training-based unsupervised domain adaption setting, we systematically explored the fast calculation for a non-negative linear, convex and concave function of ground metric.

We empirically validate its effectiveness and generality on multiple challenging benchmarks with different backbone models and achieve promising performance.

II. Related Works

Semantic segmentation predict a precise description of the class, location and shape [4]. The progress of deep learning [37], [40], [42], [36], [29], [39], [35] also contribute to a revolution semantic segmentation. [44] developed a fully convolutional network for pixel-wise or superpixel-wise classification. The conventional methods usually adopt CE loss, which equally evaluates the errors incurred by all image pixels/classes without considering the different severity-level of different mistakes [33], [7].

The importance-aware methods [8], [33] argue that the difference between object/pixel importance should be taken into account. The classes in Cityscapes are grouped as:
Group 4[most important]=$\{\mathrm{Person}, \mathrm{Car}, \mathrm{Truck}, \mathrm{Bus}, \cdots\}$;
Group 3=$\{\mathrm{Road}, \mathrm{Sidewalks}, \mathrm{Train}\}$;
Group 2=$\{\mathrm{Building}, \mathrm{Wall}, \mathrm{Fence}, \mathrm{Vegetation}, \mathrm{Terrain}\}$;
Group 1[least important]=$\{\mathrm{Sky}\}$.

The more important group will be given larger weights to compute the sum of loss in all pixels. Therefore, the misclassification of a pixel with 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 ground truth perspective rather than prediction classes. Recognizing a car to bus or road still receive the same loss is not sufficient for reliable autonomous driving. Besides, grouping manipulation is only based on human knowledge, which may differ from the way that machine perceives the world. Actually, this setting can be a special (but inferior) case of our framework.

Recently several powerful segmentation nets [10], [53] and the pose-processing strategies have also been developed to improve the initial results [27]. We note that this progress is orthogonal to our method and they can simply be added to each other.

From the loss function perspective, the focal loss [26] is developed to balance the label distribution. [24] assign different pixel with different importance. [5] proposed a tractable surrogate for the optimization of the IoU measure. [69] propose to improve the semantic segmentation performance via video propagation and label relaxation.

Wasserstein distance is a measure of distribution divergence [23]. The Wasserstein distance or optimal transportation distance has attracted the attention of adversarial generative models [3]. However, the computing cost to solve the exact distance can be a large burden. Therefore, Wasserstein distance is usually hard to be used as the loss function. Several methods propose to approximate the Wasserstein distance, which has the complexity of $\mathcal{O}(N^2)$ [13]. [16] propose to use it for the multi-class multi-label task with a linear model. Based on the previous Wasserstein loss works [18], [32], [41], [43], we propose to adapt this idea to the severity-aware segmentation.

Reinforcement learning (RL) proposes to train an RL agent to play with a dynamic environment. The optimization objective of RL is to maximize its accumulated reward. The recent developed deep RL achieved the human-level performance in many Atari Games [46].

End-to-end vision-based autonomous driving models [14] trained by RL usually have a high computational cost. [45] propose using variational inference to estimate policy parameters, while simultaneously uncovering a low dimensional latent space of actors. Similarly, [17] analyze the utility of

hierarchical representations for reuse in related tasks while learning latent space policies for RL. Recently, several works are proposed to combine semantic segmentation and policy learning [47], [48], [57], [67]. We propose that the bottleneck of segmenter can be a natural representative lower-dimensional latent space which can efficiently shrink the state space and requires fewer actor parameters. Besides, we incorporate the RL in an alternative optimization framework to learn the optimal ground matrix in a simulator with a certain reward rule.

The CARLA simulator is a realistic environment for autonomous driving. Recently, many works are implemented on CARLA [66], [9], [59], [49], [51]. However, to the best of our knowledge, this is the first effort to define the inter-class correlations in CARLA.

III. Methodology

We target to learn a segmenter h_θ, parameterized by θ, with an autoencoder structure. It projects a street view image $\mathbf{X} \in \mathbb{R}^{H_x \times W_x \times 3}$ to a prediction of semantic segmentation map $\mathbf{S} \in \mathbb{R}^{H_s \times W_s \times N}$, where N is the number of pre-defined classes in a segmentation dataset. We note the spatial size of input $H_x \times W_x$ and output $H_s \times W_s$ are not necessary the same or even have the shape of square in many segmenters. Let $\mathbf{s} = \{s_i\}_{i=1}^{N}$ be the prediction of a pixel in $h_\theta(\mathbf{X})$, *i.e.*, softmax normalized N classes probability. $i \in \{1, \cdots, N\}$ be the index of dimension (class). We perform learning over a hypothesis space $\mathcal{H}$ of h_θ. Given $\mathbf{X}$ and its target one-hot ground truth label $\mathbf{T} \in \mathbb{R}^{H_s \times W_s \times N}$, typically, learning is performed via empirical risk minimization to solve $\min_{h_\theta \in \mathcal{H}} \mathcal{L}(h_\theta(\mathbf{X}), \mathbf{T})$, with a loss $\mathcal{L}(\cdot, \cdot)$ acting as a surrogate of performance measure. Following the previous segmentation works, we define the loss for each point. But we calculate the point-wise average of a mini-batch of images to update the networks.

Unfortunately, cross-entropy (CE)-based loss treat the output dimensions independently [16], ignoring the misclassification severity on label space.

Let us define $\mathbf{t} = \{t_j\}_{j=1}^{N}$ as the target histogram distribution label that can be either one-hot or non-one-hot vector. We assume the class label possesses a ground metric $\mathbf{D}_{i,j}$, which measures the severity of misclassifying i-th class pixel into j-th class. There are N^2 possible $\mathbf{D}_{i,j}$ in a N class dataset and form a ground matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ [56]. When $\mathbf{s}$ and $\mathbf{t}$ are both histograms, the discrete measure of exact Wasserstein loss is defined as

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

where $\mathbf{W}$ is the transportation matrix with $\mathbf{W}_{i,j}$ indicating the mass moved from the i^{th} point in source distribution to the j^{th} target position. A valid transportation matrix $\mathbf{W}$ satisfies: $\mathbf{W}_{i,j} \geq 0$; $\sum_{j=0}^{N-1} \mathbf{W}_{i,j} \leq s_i$; $\sum_{i=0}^{N-1} \mathbf{W}_{i,j} \leq t_j$; $\sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \mathbf{W}_{i,j} = \min(\sum_{i=0}^{N-1} s_i, \sum_{j=0}^{N-1} t_j)$.

A possible ground matrix $\mathbf{D}$ in our application is shown in Fig. 2. For instance, classifying the car to the road ($d_{2,5}$) has a larger ground metric than car to bus ($d_{2,4}$).

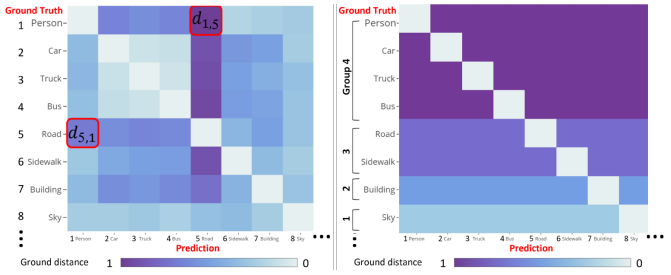

Fig. 2: Left: a possible ground matrix for severity-aware segmentation. Right: the ground matrix as an alternative for importance-aware setting.

The Wasserstein distance is identical to the Earth mover's distance when the two distributions have the same total masses (*i.e.*, $\sum_{i=0}^{N-1} s_i = \sum_{j=0}^{N-1} t_j$) and using the symmetric distance $d_{i,j}$ as $\mathbf{D}_{i,j}$. However, this is not true for our case. The entries in matrix $\mathbf{D}$ are not symmetric with respect to the main diagonal. For example, classifying the person to the road can be much severe than classifying the road to the person. Therefore, in Fig. 2, $d_{1,4}$ should have a larger value than $d_{4,1}$. We note that the importance-aware learning can be achieved by configuring the ground matrix as Fig. 2, which does not discriminate the different mistakes, *e.g.*, classifying the car into any other classes has the same punishment. The groups also just pre-defined by human but not necessarily appropriate for practical driving system.

Actually, the simple version of IAL loss propose to assign a larger weight to the pixel position that has the ground truth label with more important level, i.e., $j*$ $\in$level 3 or 4. Considering that with the one-hot label encoding, the cross-entropy loss of each pixel is -logs_{j*}. Therefore, the IAL loss can be formulated as:w_{j*}-logs_{j*}, where w_{j*} is the corresponding weight of importance level of this pixel. Since -log function is a deterministic function and is used for curving s_{j*}, the learning objective can be simplified as maximizing $w_{j*}s_{j*}$ to w_{j*}. Since $\sum s_j = 1$, it is minimizing the $\sum_j w_{j*}s_j$ for $j \neq j*$. Our proposed Wasserstein loss can be $\sum_{i=0}^{N-1} f(d_{i,j*})s_i$. When we set $f(d_{i,j*}) = w_{j*}$ for $i \neq j*$ and $f(d_{i,j*}) = 0$ for $i = j*$. The two losses are identical to each other.

A. Wasserstein training with one-hot target

The one-hot target vector $\mathbf{t}$ is the typical label for multi-class one-label dataset. We use j to index the element of $\mathbf{t}$, and j^* indicates the ground truth class. [‡], and 0 otherwise.

Theorem 1. *Assume that* $\sum_{j=0}^{N-1} t_j = \sum_{i=0}^{N-1} s_i$, *and* $\mathbf{t}$ *is a one-hot distribution with* $t_{j^*} = 1$(or $\sum_{i=0}^{N-1} s_i$)[§], *there is only one feasible optimal transport plan.*

According to the criteria of $\mathbf{W}$, all masses have to be transferred to the cluster of the ground truth label j^*, as

[‡]We use i, j interlaced for $\mathbf{s}$ and $\mathbf{t}$, since they index the same group of positions in a circle.

[§]We note that softmax cannot strictly guarantee the sum of its outputs to be 1 considering the rounding operation. However, the difference of setting t_{j^*} to 1 or $\sum_{i=0}^{N-1} s_i$) is not significant in our experiments using the typical format of softmax output which is accurate to 8 decimal places.

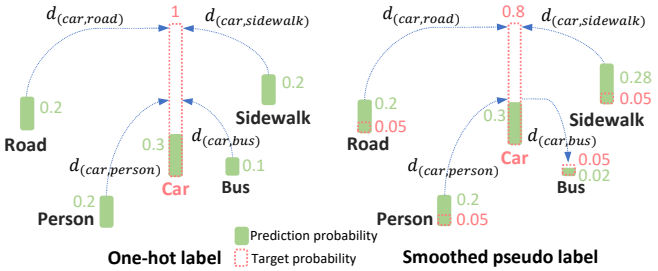

Fig. 3: Left: The only possible transport plan in one-hot target case. Right: the transportation in smoothed pseudo label $\bar{\mathbf{t}}$ is more complicated, $e.g.$, car→bus.

illustrated in Fig. 3. Then, the Wasserstein distance between softmax prediction $\mathbf{s}$ and one-hot target $\mathbf{t}$ degenerates to

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

We propose to extend the ground metric in $\mathbf{D}_{i,j}$ as $f(d_{i,j})$, where f can be a linear or increasing function proper, $e.g.$, p^{th} power of $d_{i,j}$ and Huber function. The exact solution of Eq. (2) can be computed with a complexity of $\mathcal{O}(N)$. The ground metric term $f(d_{i,j^*})$ works as the weights $w.r.t.$ s_i, which takes all classes into account following a soft attention scheme [40]. It explicitly encourages the probabilities distributing on the neighboring classes of j^*.

In contrast, the CE loss in one-hot setting can be formulated as $-1\log s_{j^*}$. Similar to the hard prediction scheme, only a single class prediction is considered resulting in a large information loss [40]. Besides, the regression loss with softmax prediction could be $f(d_{i^*,j^*})$, where i^* is the class with maximum prediction probability.

B. Monotonic increasing f w.r.t. $d_{i,j}$ as ground metric

Practically, f in $\mathbf{D}^f_{i,j} = f(d_{i,j})$ can be a positive increasing mapping function $w.r.t.$ $d_{i,j}$ for better optimization. Although the linear function is satisfactory for comparing the similarity of SIFT or hue [55], which do not involve neural network optimization.

• **Convex function** $w.r.t.$ $d_{i,j}$ **as the ground metric.** We can extend the ground metric as a nonnegative increasing and convex function of $d_{i,j}$. Here, we give some measures[¶] using the typical convex ground metric function.

$\mathcal{L}_{\mathbf{D}^\rho_{i,j}}(\mathbf{s},\mathbf{t})$, the Wasserstein measure using d^ρ as the ground metric with $\rho = 2, 3, \cdots$. The case $\rho = 2$ is equivalent to the Cramér distance [52]. Note that the Cramér distance is not a distance metric proper. However, its square root is.

$$\mathbf{D}^\rho_{i,j} = d^\rho_{i,j} \tag{3}$$

$\mathcal{L}_{\mathbf{D}^{H_\tau}_{i,j}}(\mathbf{s},\mathbf{t})$, the Wasserstein measure using a Huber cost function with a parameter τ.

$$\mathbf{D}^{H_\tau}_{i,j} = \begin{cases} d^2_{i,j} & \text{if } d_{i,j} \le \tau \\ \tau(2d_{i,j} - \tau) & \text{otherwise.} \end{cases} \tag{4}$$

[¶]We refer to "measure", since a ρ^{th}-root normalization is required to get a distance [60], which satisfies three properties: positive definiteness, symmetry and triangle inequality.

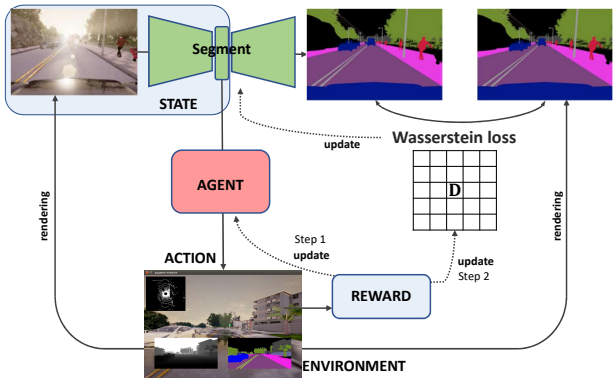

Fig. 4: The reinforced alternative optimization framework to learn actor-critic agent and ground matrix simultaneously.

• **Concave function** $w.r.t.$ $d_{i,j}$ **as the ground metric.** In practice, it may be not meaningful to set the ground metric as a nonnegative, concave and increasing function $w.r.t.$ $d_{i,j}$. We note that the computation speed of exact solution in conservative target label case is usually not satisfactory, but the step function $f(t) = 1_{t\neq 0}$ (one everywhere except at 0) can be a special case, which has significantly less complexity [60]. Assuming that the $f(t) = 1_{t\neq 0}$, the Wasserstein metric between two normalized discrete histograms on N bins is simplified to the ℓ_1 distance.

$$\mathcal{L}_{1_{d_{i,j}\neq 0}}(\mathbf{s},\mathbf{t}) = \frac{1}{2}\sum_{i=0}^{N-1} |\mathrm{s}_i - \mathrm{t}_i| = \frac{1}{2}||\mathbf{s}-\mathbf{t}||_1 \tag{5}$$

where $||\cdot||_1$ is the discrete ℓ_1 norm. Unfortunately, its fast computation is at the cost of losing the ability to discriminate the difference of probability in different bins.

C. Learn severity-aware ground matrix

Other than the pre-defined ground matrix, we further propose to learn the ground matrix in a simulator with our autonomous driving agent following the alternative optimization.

The overall framework is illustrated in Fig. 4. We choose a high-reality simulator, the CARLA [14], as our environment. The view of a monocular camera placed at the front the car is rendered as $\mathbf{X}$. Segmenter takes $\mathbf{X}$ as input and predicts the segmentation image $\mathbf{S}$ which is compared with target $\mathbf{T}$ with Wasserstein loss.

An agent learns to interact with the environment following a partially observable Markov decision process (POMDP). For the time step t, a RL agent observes the state s_t in a state space $\mathcal{S}$ and predict an action a_t from an action space $\mathcal{A}$, following the RL policy $\pi(a_t|s_t)$, which is the behavior of the agent. Then, the action will result in the change of environment and move to the next state s_{t+1}, and receive a reward $r_t(s_t, a_t) \in \mathcal{R} \subseteq \mathbb{R}$ from the dynamic environment. The optimization objective of an optimal policy π^* is to maximize the discounted total return $R_t = \sum_{i\geq 0}^{T} \gamma^i r_{t+i}(s_t, a_t)$ in expectation, where $\gamma \in [0, 1)$ is used to balance the current and the long-term rewards [25].

Instead of using $\mathbf{X}$ as our state [14], we propose to utilize the latent representation of our segmenter. It can be either feature

vector or feature maps according to the backbone. [14] 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, segmentation map or its latent representation has sufficient information ($e.g.$, each object and their and precise location) to guide the driving, and is robust to appearance variation ($e.g.$, weather, lighting). Since a high proportion of pixels have the same label as their neighbors in **S**, there are a large of room to reduce its redundancy.

The network takes two latent representations as input, which is the two most recent at this step, as well as a vector of sensor readings. The two inputs are feed to two different branches: feature maps by a convolutional module, measurements by a fully-connected network. The two branches are merged later and further process the fused information.

In the context of autonomous driving, we define the action 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 $o_l, o_r \in [0, 1]$ measure the degree of off-line or off-road respectively, and $c \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$ indicates there is no/S0/S1/S2/S3 level crash, where S0, S1, S2, S3 denotes the severity is negligible/minor, major, hazardous and catastrophic defined in [ISO26262] [20]. α, β and ψ are a set of positive weights to balance the punishments, we empirically set $\alpha = 1, \beta = 1$ and $\psi = 10$ in all of our experiences. 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.

Given a continuous action space, the value-based RL, for example the Q-Learning, cannot be able to pedict continuous values. Therefore, we resort to the actor-critic algorithm. As a kind of the policy-based method, the objective of RL is to learn a policy $\pi_\theta(a_t|s_t)$ to maximize the expected reward $J(\theta)$ over all possible decisions. With the policy gradient theorem [58], the gradient of the parameters given the objective function has the form:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t)(Q(s_t, a_t) - b(s_t))] \quad (6)$$

where $Q(s_t, a_t) = \mathbb{E}[R_t|s_t, a_t]$ can be defined as the state-action value function. We note that the initial action a_t is provided to calculate the expected return when starting in the state s_t. Moreover, the baseline function $b(s_t)$ is usually subtracted to reduce the variance and not changing the estimated gradient [62], [2]. A possible baseline function can be the state only value function $V(s_t) = \mathbb{E}[R_t|s_t]$. It is similar to $Q(s_t, a_t)$, except the a_t is not given here. The advantage function is defined as $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ [25]. Eq.(4) then becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)] \quad (7)$$

It can be regarded as a specific case of actor-critic RL, in which $\pi_\theta(a_t|s_t)$ can be the actor and the $A(s_t, a_t)$ can be the critic. To reduce the number of required parameters, the parameterized temporal difference error $\delta_\omega = r_t + \gamma V_\omega(S_{s+1}) - V_\omega(S_s)$ can be used to approximate the advantage function. We adopt two different symbols θ and ω to denote the actor and critic function respectively. We note that the most of these parameters are shared in a mainstream neural network, then separated to two branches for policy and value predictions. We further adapt the A3C to its off-policy version to stabilize and speed up our training.

After configuring our RL module, we propose to adaptively learn the ground metric along with the training of actor following the alternative optimization.

Step 1: Fixing the ground matrix to compute $\mathcal{L}_{\mathbf{D}_{i,j}}(\mathbf{s}, \mathbf{t})$ and updating the network parameters of our actor-critic module.

Step 2: Fixing the network parameters and postprocessing the ground matrix with the feature-level ℓ_1 distances between different classes.

In this round, we use the normalized second-to-last convolutional layer's channel-wise response at each point as a feature vector, since there is no subsequent non-linearities. Therefore, it is meaningful to average the feature vectors in each position that corresponds to the pixel in image-level with the same class label to compute their centroid and reconstruct $\mathbf{D}_{i,j}$ using the ℓ_1 distances between these centroids $\overline{d}_{i,j}$. To avoid the model collapse, we construct the $\mathbf{D}_{i,j} = \frac{1}{1+\alpha}\left\{f(\overline{d}_{i,j}) + \alpha f(d_{i,j})\right\}$ in each round, and decrease α from 10 to 0 gradually in the training.

IV. Experiments

In the experiment section, we provide the implementation details and experimental results on two typical autonomous driving benchmarks ($i.e.$, Cityscapes [12] and CamVid [6]) and the CARLA simulator [14]. Other than the comparisons, We also give the detailed ablation study to illustrate the effectiveness of each module and their combinations. Our Wasserstein loss framework is implemented in PyTorch platform. All of the networks are pre-trained with CE loss as their vanilla version.

We follow the RL agent structure proposed in [61]. The two most recent latent feature maps observed by the agent and a vector of measurements are feed into the two branches of the agent. The measurement vector includes the current speed of the car, distance to the goal, damage from collisions, and the current high-level command provided by the topological planner, in one-hot encoding. We note that the inputs are processed by two separate branches. Specifically, the feature maps is feed to a convolutional branch, while the measurements are feed to a fully-connected branch. After the processing, we concatenate the two outputs to fuse the information.

Our RL framework is trained with 10 parallel actor threads, for a total of 10 million environment steps. As in previous work [21], we also choose 20-step rollouts. The initial learning rate is set to 0.0007, and with the entropy regularization of 0.01. Along with the training, the learning rate our network is linearly decreased to zero. We note that the Wasserstein loss is defined for each pixel in the image, but we calculate the point-wise average of a mini-batch of images to update the networks.

According to CARLA simulator [14], the inputs are the camera image and the sensor reading information. We adopt

	Group4							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
$+\mathcal{L}_{d_{i,j}}$	86.4	48.7	92.8	78.5	68.2	40.2	62.8	67.4
$+\mathcal{L}_{\mathbf{D}^2_{i,j}}$	87.5	**50.2**	**93.4**	**79.8**	69.5	**42.0**	64.3	**68.0**
$+\mathcal{L}_{\mathbf{D}^{H^\tau}_{i,j}}$	**87.6**	49.8	93.2	79.5	**70.3**	41.6	63.6	67.9
$+\mathcal{L}_1$	63.0	41.5	87.4	40.1	43.7	38.2	50.6	56.3
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
$+\mathcal{L}_{d_{i,j}}$	90.7	48.7	95.5	70.8	75.3	46.2	73.3	69.1
$+\mathcal{L}_{\mathbf{D}^2_{i,j}}$	90.9	**49.6**	**96.8**	71.4	77.6	**46.3**	**75.1**	69.3
$+\mathcal{L}_{\mathbf{D}^{H^\tau}_{i,j}}$	**90.1**	49.5	**96.8**	72.6	**77.8**	46.2	75.0	**69.5**
$+\mathcal{L}_1$	72.5	40.3	85.2	39.4	48.7	41.0	52.9	59.1
FCN	75.4	50.5	91.9	35.3	49.1	50.7	65.2	64.3
+IAL	90.4	56.6	93.7	68.5	74.6	31.5	**81.5**	71.9
$+\mathcal{L}_{d_{i,j}}$	89.5	60.3	92.5	73.2	73.5	54.2	71.0	71.7
$+\mathcal{L}_{\mathbf{D}^2_{i,j}}$	90.6	56.5	93.8	**74.6**	74.4	**56.1**	70.3	72.0
$+\mathcal{L}_{\mathbf{D}^{H^\tau}_{i,j}}$	**91.5**	**59.4**	**95.2**	74.3	**74.6**	52.4	72.4	**72.2**
$+\mathcal{L}_1$	78.3	60.1	88.4	49.5	52.2	51.6	69.1	65.2

TABLE I: The comparison results of various methods of Cityscapes Group 4 with SegNet, ENet and FCN backbone.

two fully connected (FC) layers (64,64) to process the vector of sensor reading. We apply two convolutional layers with 3×3×32 and 3×3×16 kernels and 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. 5, our actor-critic uses two fully connected (FC) layers (256,128) then cascade two sub-branches with two fully connected layers (64,16). The number of the output unit is set as 3 which indicates the steering, throttle and brake.

The evaluation using third-party reinforcement framework on CARLA follows the experiment setting, network structures and hyperparameter settings as [15][‖].

We introduce the used evaluation metrics as follow.

• The intersection-over-union (IoU) is defined as:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{8}$$

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.

• Metrics used in third-party evaluation [15]

The metrics used in the third-party evaluation in CARLA ($i.e.$, Table 5) are as follows:

Drive% measures the number of steps that took place during the evaluation divided by 720,000. A value of 100% indicates the agent does not have the early termination. In contrast, a low value indicates the failures.

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

	Group3			Group4			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
$+\mathcal{L}_{d_{i,j}}$	98.5	93.2	28.3	87.4	71.3	60.0	72.4
$+\mathcal{L}_{\mathbf{D}^2_{i,j}}$	**98.7**	94.6	**29.7**	89.5	73.4	**60.7**	**72.8**
$+\mathcal{L}_{\mathbf{D}^{H^\tau}_{i,j}}$	98.5	**95.0**	29.5	**89.7**	**73.5**	60.6	**72.8**

TABLE II: The comparison results of various methods on the Group 3/4 of CamVid dataset using FCN as backbone.

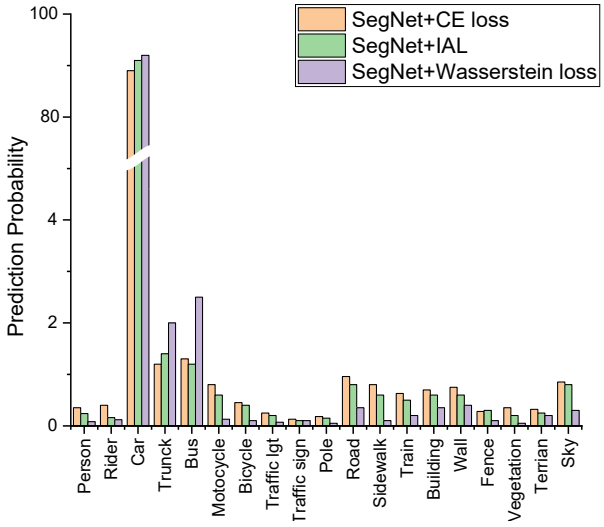

Fig. 5: The confusion statistics of classifying car on the testing set of Cityscapes dataset with SegNet backbone.

Kmis the total kilometers driven across all steps. It is also the function of mean speed and drive%.

Km/Hr denotes the mean speed taken across all steps. The pre-set maximum speed in CARLA is 25km/hr.

Km/OOL denotes the driving distance on average between each out of lane (OOL) instances. Ideally, this value is infinite if there is no OOL infraction, the value can be infinite.

Km/Collision denotes the driving distance on average between each collision with an object in the environment. Ideally, this value is infinite if no collisions occur.

A. Importance-aware SS with one-hot label

We firstly pre-define our ground matrix as Fig. 2 right to achieve the importance-aware SS. Following the setting in IAL [8], [7], we choose the SegNet [4] and ENet [50] as our backbone to fairly compair with IAL. We note that our method can be applied on more advanced backbone [53]. The conventional CE loss in their vanilla version is replaced by IAL and our Wasserstein loss.

The recent Cityscapes dataset has 2975/500/1525 images for training/validation/testing respectively. The 19 most frequently used classes are chosen and grouped as IAL. Table I shows that the class in group 4 are segmented with higher IoU when considering the importance of each class. Our Wasserstein loss usually outperforms IAL by more than 2%, especially apply the convex function $w.r.t.d_{i,j}$. The improvements $w.r.t.$ Motor are more than 15% over IAL.

	Group4							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
$\mathcal{L}_{d_{i,j}}$	65.4	33.7	88.5	36.2	44.8	39.3	48.4	47.8
$\mathcal{L}_{\mathbf{D}^2_{i,j}}$	65.7	34.0	88.9	36.7	45.3	39.6	49.1	48.0
$\mathcal{L}_{\mathbf{D}^{H\tau}_{i,j}}$	**66.2**	**34.7**	**89.5**	**37.1**	**46.0**	**40.8**	**50.5**	**48.3**

TABLE III: The comparison results of various methods on the Group4 of GTA5→Cityscapes unsupervised domain adaptation using DeeplabV2 as backbone.

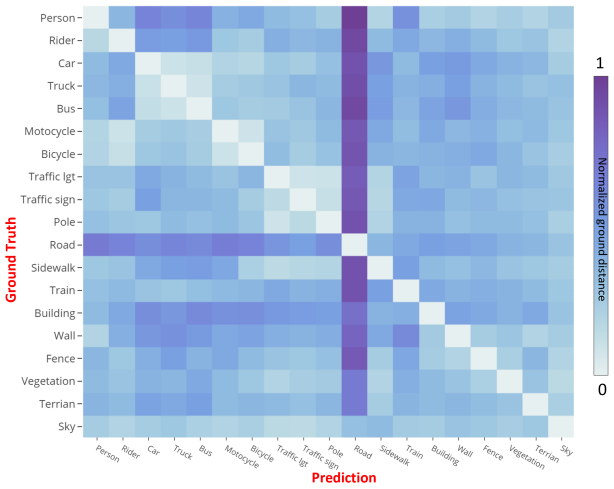

Fig. 6: Normalized adaptively learned ground matrix in CARLA simulator with ENet backbone.

The CamVid dataset contains 367, 26 and 233 images for training, validation, and testing respectively. To make fair evaluation, we choose the same setting and measurements as IAL, and report the results in Table II. We note that fine-tuning a public available trained FCN segmenter [44] with Wasserstein loss is 1.5× faster than the training of IAL. We note that the training use only Wasserstein loss can be 2.2 or 2.4 times slower than CE loss in Cityscapes or CamVid datasets respectively. We have added the related comparison in our revised version. Although the IoU of some unimportant classes may drop, this will have a limited impact on driving safety. We note that the mean IoU of all classes can still be comparable or improved since we introduced a more strict objective than CE loss only. Since the metrics used in IAL cannot evidence the superiority of severity-aware setting, we give additional confusion statistics in figure 5.

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

B. Wasserstein training with conservative target

The self-training scheme [70] can be a promising solution for the unsupervised domain adaptation in both classification and semantic segmentation [54], which involves an iterative process. Specifically, it first predict on the target domain and then taking the confident predictions as pseudo-labels for retraining. Unavoidable, the pseudo-labels can be noisy and unreliable. In consequence, the self-training can put over-confident label belief on wrong classes, leading to deviated solutions with propagated errors. [70] propose to construct the smoothed pseudo-label $\bar{\mathbf{t}}$, which smooth the one-hot pseudo-label to a conservative (i.e., non one-hot) target distribution. Using the conservative distribution as the label, the fast computing of Wasserstein distance in Eq. (2) is not applicable.

The closed-form result of the general Wasserstein distance can have the complexity higher than $\mathcal{O}(N^3)$, which cannot satisfy the speed requirement of the loss function. Therefore, a possible solution is to approximate the Wasserstein distance, which usually has the complexity of $\mathcal{O}(N^2)$. [13] proposes an efficient approximation of both the transport matrix in and the subgradient of the loss, which is essentially a matrix balancing problem that has been well-studied in numerical linear algebra [22].

C. Importance-aware SS with conservative label

We further test our method for unsupervised domain adaptation with constrained self-training, i.e., label entropy regularizer (LRENT) [70]. We compute the approximate Wasserstein distance as the loss. Table III shows the performance of GTA5→Cityscapes adaptation and outperforms the CE loss-based LRENT by more than 5% in these important classes consistently. The improvements of $\mathcal{L}_{\mathbf{D}^{H\tau}_{i,j}}$ over $\mathcal{L}_{\mathbf{D}^2_{i,j}}$ are more significant than the one-hot case. This is probably because that the Huber function is more robust to the label noise which is common for the pseudo label in self-learning method. This task also indicates that our method can be a general alternative objective of CE loss and be applied in a plug and play fashion. We note that using Eq. 6, the Wasserstein loss will totally lost the discriminability of different missclassification.

D. Severity-aware SS with 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-define a severity-aware ground matrix with human knowledge, we propose to learn it in the CARLA simulator** and show our result with ENet backbone in Fig. 6. We train our actor-critic with 10 parallel actor threads as [14] for a total of 5-million steps. 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 [14] takes 12 days to train a reinforcement learning framework. The time cost will be intractable when we incorporate a ground matrix simultaneously. To evidence the effectiveness of our method, we show a segmentation example in Fig. 7.

**https://carla.org

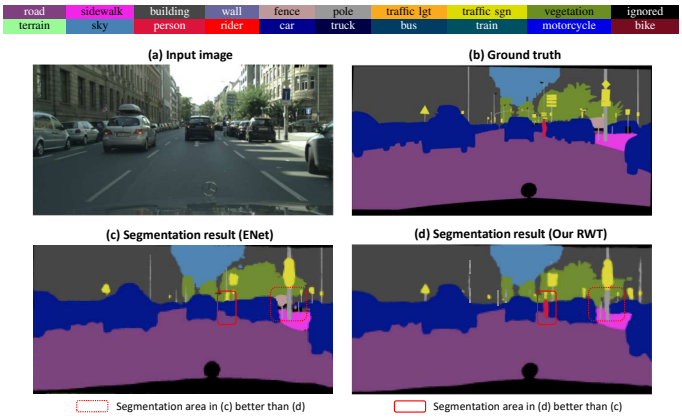

Fig. 7: Representative semantic segmentation result of ENet and our reinforcement Wasserstein training with ENet backbone on Cityscapes dataset. The two image has the same mIoU but the misclassification of the person may lead to more severity result.

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 IV: The average distance (km) between the two infractions of using the ENet trained only with CE loss (wo/) or fine-tuned with Wasserstein loss (w/) in our reinforcement learning framework. Higher is better.

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 V: Results of different training methods using Deeplab backbone and Deeplab/[15] evaluation on the CARLA simulator. Higher is better.

Besides,the Wasserstein loss is stabilized after 3×10^5 steps. Training with more steps does not affect the performance until 5×10^5 steps. Actually, based on our experiments for 10×10^5steps, the curve can be stable. The window for training step of Wasserstein loss does not require careful tuning in our tested datasets.

CARLA characterizes the approaches by average distance traveled between infractions of the following five types: opposite lane, Sidewalk, collision with static object, collision with car, collision with person.

CARLA offers a fine-grained evaluation of driving polices which characterize the approaches by the average distance between different collisions and more than 30% off-line or off-road. The results are reported in Table IV. Rather than test 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 misclassification of person, the average distance between two collisions with a person almost doubled in all of the testing cases. Besides, in VI, we evaluate the success rate [11] of different training methods using Deeplab backbone and 10 hours of demonstration on the regular traffic CARLA simulator. We can see that the Wasserstein loss can consistently improve the success rate.

Method	Training	New town	New Weather
Deeplab wo/	58.2	33.7	30.5
Deeplab w/ IAL	62.5	38.3	35.2
Deeplab w/ A-$\mathcal{L}_{d_{i,j}}$	65.7	41.6	40.3

TABLE VI: Success rate of different training methods using Deeplab backbone and 10 hours of demonstration on the regular traffic CARLA simulator.

Other than using our reinforcement learning framework to make the driving decision, we also evaluate our segmented results using an independent autonomous driving system. [15] propose to process the front view image in CARLA with Deeplab [10] 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 Wasserstein loss or IAL. Following the experiment setting and evaluation metrics, we give the comparison in Table V. We use the prefix A to denote the adaptive ground metric learning. The improvements over Deeplab and IAL trained Deeplab indicate that our segmenter can offer more reliable and safe segmentation results for the driving system.

V. Conclusions

In this paper, we proposed a concise loss function for semantic segmentation in context of safe driving, based on the Wasserstein distance. The ground metric of Wasserstein distance represents the pair-wise severity and can be either predefined or learned by alternative optimization. The importance-aware problem can be a special case of our framework. Configuring a convex function of $d_{i,j}$ can further improve its performance. 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 not only achieve the promising results in importance-aware tasks, but also improve the autonomous driving metrics in CARLA simulator significantly. For the feature work, we are planing to apply it to more advanced backbones and use real world evaluations to adjust the ground matrix [68].

VI. Acknowledgements

This work was supported by the Jangsu Youth Programme [grant number SBK2020041180], National Natural Science Foundation of China, Younth Programme [grant number 61705221], and Hong Kong Government General Research Fund GRF (Ref. No.152202/14E) are greatly appreciated.

REFERENCES

[1] J. M. Á. Alvarez and A. M. Lopez. Road detection based on illuminant invariance. *IEEE transactions on intelligent transportation systems*, 12(1):184–193, 2010.

[2] A. 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.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017.

[5] M. Berman, A. R. Triki, and M. B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, 2017.

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

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

[8] B.-k. Chen, C. Gong, and J. Yang. Importance-aware semantic segmentation for autonomous driving system. In *IJCAI*, pages 1504–1510, 2017.

[9] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. *arXiv preprint arXiv:1912.12294*, 2019.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017.

[11] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9329–9338, 2019.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[13] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[15] G. Fennessy. *Autonomous Vehicle End-to-End Reinforcement Learning Model and the Effects of Image Segmentation on Model Quality*. PhD thesis, Vanderbilt University, 2019.

[16] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

[17] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine. Latent space policies for hierarchical reinforcement learning. *arXiv:1804.02808*, 2018.

[18] Y. Han, X. Liu, Z. Sheng, Y. Ren, X. Han, J. You, R. Liu, and Z. Luo. Wasserstein loss-based deep object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[19] G. He, X. Liu, F. Fan, and J. You. Classification-aware semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[20] I. ISO. 26262: Road vehicles-functional safety. *International Standard ISO/FDIS*, 2011.

[21] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

[22] P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3):1029–1047, 2013.

[23] S. Kolouri, Y. Zou, and G. K. Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016.

[24] X. Li, Z. Liu, P. Luo, C. Change Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3193–3202, 2017.

[25] Y. Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] F. Liu, G. Lin, and C. Shen. Crf learning with cnn features for image segmentation. *Pattern Recognition*, 48(10):2983–2992, 2015.

[28] X. Liu, T. Che, Y. Lu, C. Yang, S. Li, and J. You. Auto3d: Novel view synthesis through unsupervisely learned variational viewpoint and global 3d representation. *arXiv preprint arXiv:2007.06620*, 2020.

[29] X. Liu, Y. Ge, C. Yang, and P. Jia. Adaptive metric learning with deep neural networks for video-based facial expression recognition. *Journal of Electronic Imaging*, 27(1):013022, 2018.

[30] X. Liu, Z. Guo, S. Li, L. Kong, P. Jia, J. You, and B. Kumar. Permutation-invariant feature restructuring for correlation-aware image set-based recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4986–4996, 2019.

[31] X. Liu, Z. Guo, J. You, and B. V. Kumar. Dependency-aware attention control for image set-based face recognition. *IEEE Transactions on Information Forensics and Security*, 15:1501–1512, 2019.

[32] X. Liu, X. Han, Y. Qiao, Y. Ge, S. Li, and J. Lu. Unimodal-uniform constrained wasserstein training for medical diagnosis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[33] X. Liu, Y. Han, S. Bai, Y. Ge, T. Wang, X. Han, S. Li, J. You, and J. Lu. Importance-aware semantic segmentation in self-driving with discrete wasserstein training. 2020.

[34] X. Liu, W. Ji, J. You, G. E. Fakhri, and J. Woo. Severity-aware semantic segmentation with reinforced wasserstein training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12566–12575, 2020.

[35] X. Liu, L. Kong, Z. Diao, and P. Jia. Line-scan system for continuous hand authentication. *Optical Engineering*, 56(3):033106, 2017.

[36] X. Liu, B. V. Kumar, Y. Ge, C. Yang, J. You, and P. Jia. Normalized face image generation with perceptron generative adversarial networks. In *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–8. IEEE, 2018.

[37] X. Liu, B. V. Kumar, P. Jia, and J. You. Hard negative generation for identity-disentangled facial expression recognition. *Pattern Recognition*, 88:1–12, 2019.

[38] X. Liu, S. Li, L. Kong, W. Xie, P. Jia, J. You, and B. 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.

[39] X. Liu, Z. Li, L. Kong, Z. Diao, J. Yan, Y. Zou, C. Yang, P. Jia, and J. 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. IEEE, 2018.

[40] X. Liu, B. Vijaya Kumar, C. Yang, Q. Tang, and J. 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.

[41] X. Liu, Y. Zou, T. Che, P. Ding, P. Jia, J. You, and B. Kumar. Conservative wasserstein training for pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8262–8272, 2019.

[42] X. Liu, Y. Zou, L. Kong, Z. Diao, J. Yan, J. Wang, S. Li, P. Jia, and J. You. Data augmentation via latent space interpolation for image classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 728–733. IEEE, 2018.

[43] X. Liu, Y. Zou, Y. Song, C. Yang, J. You, and B. 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.

[44] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[45] K. S. Luck, J. Pajarinen, E. Berger, V. Kyrki, and H. B. Amor. Sparse latent space policy search. In *AAAI*, 2016.

[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[47] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019.

[48] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018.

[49] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger. Learning

situational driving. In *CVPR*, volume 2, page 8, 2020.
[50] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *ICLR*, 2017.
[51] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. CVPR, 2020.
[52] M. L. Rizzo and G. J. Székely. Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38, 2016.
[53] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2019.
[54] E. Romera, L. M. Bergasa, K. Yang, J. M. Alvarez, and R. Barea. Bridging the day and night domain gap for semantic segmentation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1312–1318. IEEE, 2019.
[55] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.
[56] L. Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129, 1985.
[57] A. Sax, J. O. Zhang, B. Emi, A. Zamir, S. Savarese, L. Guibas, and J. Malik. Learning to navigate using mid-level visual priors. *arXiv preprint arXiv:1912.11121*, 2019.
[58] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 2000.
[59] M. Toromanoff, E. Wirbel, and F. Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. *arXiv preprint arXiv:1911.10868*, 2019.
[60] C. Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.
[61] M. Volodymyr, K. Koray, S. David, A. R. Andrei, and V. Joel. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
[62] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.
[63] K. Xiang, K. Wang, and K. Yang. A comparative study of high-recall real-time semantic segmentation based on swift factorized network. In *Artificial Intelligence and Machine Learning in Defense Applications*, volume 11169, page 111690C. International Society for Optics and Photonics, 2019.
[64] K. Xiang, K. Wang, and K. Yang. Importance-aware semantic segmentation with efficient pyramidal context network for navigational assistant systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3412–3418. IEEE, 2019.
[65] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang. Pass: Panoramic annular semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2019.
[66] A. Zhao, T. He, Y. Liang, H. Huang, G. V. d. Broeck, and S. Soatto. Lates: Latent space distillation for teacher-student driving policy learning. *arXiv preprint arXiv:1912.02973*, 2019.
[67] B. Zhou, P. Krähenbühl, and V. Koltun. Does computer vision matter for action? *arXiv preprint arXiv:1905.12887*, 2019.
[68] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot. Automated evaluation of semantic segmentation robustness for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2019.
[69] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.
[70] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. Confidence regularized self-training. *ICCV*, 2019.