Lookahead Adversarial Semantic Segmentation

Hadi Jamali-Rad [1] **Attila Szabó** [1 2] **Matteo Presutto** [1]

Abstract

Adversarial learning is shown to be an effective approach for improving semantic segmentation quality by enforcing higher-level pixel correlations. However, state-of-the-art semantic segmentation models cannot be easily plugged into an adversarial setting because they are not designed to accommodate convergence and stability issues in adversarial networks. To address this, we introduce a novel lookahead adversarial learning approach (LoAd) with an embedded label map aggregation module. We show that the proposed solution can alleviate divergence issues in an adversarial semantic segmentation setting and results in considerable performance improvements (up to 5% in some classes) on two standard datasets.

1. Introduction

Semantic segmentation is one of the most fundamental problems in computer vision. It is a pivotal step towards content-based image analysis and scene understanding as it empowers machines to distinguish between different regions of an image based on its semantic context. Semantic segmentation has received an upsurge of attention recently owing to its wide variety of applications in medical imaging (Ronneberger et al., 2015; Rezaei et al., 2017), autonomous driving (Menze & Geiger, 2015; Cordts et al., 2016), satellite image processing (Volpi & Ferrari, 2015; Henry et al., 2018), and robotics (Geiger et al., 2013; Shvets et al., 2018), to name a few. Recent advances in deep learning and convolutional neural networks (CNNs) revolutionized this fields resulting in state-of-the-art image segmentation algorithms such as FCN (Long et al., 2015), U-Net (Ronneberger et al., 2015), PSPNet (Zhao et al., 2017), EncNet (Zhang et al., 2018a), Exfuse (Zhang et al., 2018b), DeepLabv3+ (Chen et al., 2018), PS and Panoptic-DeepLab (Kirillov et al., 2019; Cheng et al., 2019), and HRNet (Wang et al., 2020).

Majority of these deep learning based methods formulate semantic segmentation as a classification problem where cross entropy (CE) with pixel independence assumption is employed as the optimization loss function. However, in practice, adjacent pixels of an image are highly correlated. These methods implicitly assume that correlation among pixels would be learned as receptive field of CNNs increases going deeper with convolutions. Recent studies challenge this assumption and propose different approaches to capture pixel inter-dependencies. For instance, CRFs can be used to model pixel relationships and enforce label consistency between pixels (Liu et al., 2017; Chen et al., 2017; Shen et al., 2017; Liu et al., 2017). However, CRFs are known to be extremely time-consuming at inference and sensitive to variations in visual appearance. An alternative approach is extracting pixel affinity information from images and fusing them back to predicted label maps (Ke et al., 2018); this comes at the cost of extra model branches and larger memory requirements. Other studies have proposed using different loss functions that encode the mutual information or structural similarity among nearby pixels in a regional fashion (Zhao et al., 2019a;b) and have shown improvements. However, these losses can be derived in a sub-optimal manner by considering a small patch of pixels.

Another avenue that has been explored to enforce structure in segmentation is employing adversarial learning (Luc et al., 2016; Souly et al., 2017; Xue et al., 2018; Hung et al., 2018). In this setup, a segmentor-discriminator pair compete to outperform each other in creating realistic label maps and distinguishing them from ground truth ones. We think a conditional adversarial approach similar to (Luc et al., 2016; Souly et al., 2017) has the capacity to capture these pixel inter-dependencies and correlations in a more general (and not only local) fashion when compared to methods proposed in (Zhao et al., 2019a;b). On the other hand, plugging state-of-the-art semantic segmentation models in an adversarial setting is prone to the well-known divergence and mode collapse issues (Arjovsky et al., 2017; Goodfellow et al., 2014b). Specific architecture designs for generator and discriminator networks can help to stabilize the setup, but at the cost of limiting the application domain of adversarial networks. Bridging the gap between employing the state-of-the-art semantic segmentation models in adversarial settings and helping to stabilize them is the core idea of the proposed *lookahead adversarial* learning (LoAd) approach.

[1]Shell Global Solutions International B.V., Amsterdam, NL [2]Universiteit van Amsterdam (UvA), Amsterdam, NL. Correspondence to: Hadi Jamali-Rad <hadi.jamali-rad@shell.com>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

2. Conditional Adversarial Training

Let $\mathcal{D}_t = \{(\mathbf{X}, \mathbf{Y})_1, ..., (\mathbf{X}, \mathbf{Y})_M\}$ be the training dataset containing M samples with $\mathcal{X}_t = \{\mathbf{X} | (\mathbf{X}, \mathbf{Y}) \in \mathcal{D}_t\}$ and $\mathcal{Y}_t = \{\mathbf{Y} | (\mathbf{X}, \mathbf{Y}) \in \mathcal{D}_t\}$ respectively denoting the set of images and their corresponding label maps. Here, $\mathbf{X}$ is of size $H \times W \times 3$ for RGB images with a total of $H \times W = N$ pixels. The corresponding label map $\mathbf{Y}$ is of size $H \times W$ with elements in $\mathcal{K} = \{1, \cdots, K\}$ where K is the number of classes in segmentation task. An adaptation of conditional generative adversarial networks (CGANs) (Goodfellow et al., 2014a; Mirza & Osindero, 2014; Isola et al., 2017) for semantic segmentation would aim at creating the most probable map $\mathbf{Y}$ per input image $\mathbf{X}$. So, we solve a two-player min-max game to estimate $P(\mathbf{Y}|\mathbf{X})$

$$\min_G \max_D \mathcal{L}(G, D) = \mathbb{E}_{\mathbf{Y} \sim P_{\mathcal{D}_t}(\mathbf{Y})} \big[\log\left(D(\mathbf{Y}|\mathbf{X})\right)\big] + \mathbb{E}_{\mathbf{Y} \sim P_g(\mathbf{Y})} \big[\log\left(1 - D(\mathbf{Y}|\mathbf{X})\right)\big], \quad (1)$$

where G denotes a generator (segmentor) parameterized with $\boldsymbol{\theta}_g$, D stands for a discriminator parameterized with $\boldsymbol{\theta}_d$, and the loss function $\mathcal{L}(G, D)$ should be minimized w.r.t. $\boldsymbol{\theta}_g$ and maximized w.r.t. $\boldsymbol{\theta}_d$. Typically, both G and D are CNN's. Several interesting studies such as (Luc et al., 2016; Souly et al., 2017; Hung et al., 2018) suggest applying a hybrid loss combining the conditional adversarial loss in (1) with a cross-entropy (CE) pixel-wise term as follows

$$\mathcal{L}_h = \sum_{m=1}^{M} \text{CE}(\mathbf{Y}_m, \hat{\mathbf{Y}}_m) + \lambda \sum_{m=1}^{M} \log\left(D(\mathbf{Y}_m|\mathbf{X}_m)\right) + \lambda \sum_{m=1}^{M} \log\left(1 - D(\hat{\mathbf{Y}}_m|\mathbf{X}_m)\right), \quad (2)$$

where (1) is simplified for a binary classification setting in which the discriminator is to decide whether a sample label map is ground truth ($\mathbf{Y} \sim P_{\mathcal{D}_t}$) or generated ($\mathbf{Y} \sim P_g$) by the segmentor. The pixel-wise loss is computed using a multi-class CE between the 1-hot encoded versions of the original label map $\mathbf{Y}$ and the inferred one $\hat{\mathbf{Y}}$ using $-\sum_{i=1}^{N}\sum_{c=1}^{K} y_{i,c} \log(\hat{y}_{i,c})$, with y_i denoting the ith element of $\mathbf{Y}$. Obviously, only the second and the third terms in (2) are relevant when training the discriminator. When training the generator, (Luc et al., 2016) proposes to keep the first and the third terms. Next, a standard gradient decent ascent (GDA) (Lin et al., 2019) is applied to the two-player min-max game.

We decided to take a different approach for two reasons. First, following the propositions in (Nouiehed et al., 2019; Ostrovskii et al., 2020), and in contrast to (Luc et al., 2016), we avoid an alternating GDA in optimizing the generator and discriminator networks. Instead, in every "cycle" of the proposed adversarial approach (Algorithm 1), we keep training the discriminator with a dynamically updated label map dataset to reach sufficient accuracy before switching back to training the generator. Second, based on our experience when incorporating state-of-the-art semantic segmentation models (like DeepLabv3+) in an adversarial setting, presence of the pixel-wise CE loss exacerbates the divergence issues. Therefore, we approach the problem in two stages as follows. Stage 1: if not pre-trained on $\mathcal{D}_t$, we first train the segmentation network using only CE pixel-wise loss up to a reasonable performance (no hard constraints). Stage 2: we then deactivate the pixel-wise CE loss (set it to 0) and run LoAd to boost the performance. At this stage, when training the discriminator both second and third terms of (2) will be active, and when training the segmentation network only the third term will be used.

Algorithm 1: LoAd for Semantic Segmentation

Initialize : $\psi = 0$, $g^s = g_0$, $\mathcal{B} = g^s(\mathcal{X}_t)$
Input: max cycle: Ψ, max patience: Γ, β_l, β_u
$\mu^s, \mu^*, \mu \leftarrow$ evaluate mIoU
`Train Discriminator`$(\mathcal{D}_t \cup \mathcal{B})$
while $\psi < \Psi$ **do**
  start a divergence patience counter: $\gamma \leftarrow 0$
  while $\mu^s - \beta_l < \mu$ and $\gamma < \Gamma$ **do**
    update model: $g \leftarrow$ `Train Advers.`
    $\mu \leftarrow$ evaluate mIoU
    $\mu^* \leftarrow$ best $\mu > \mu^s + \beta_u$
    update best model: $g^* \leftarrow g$
    $\gamma \leftarrow \gamma + 1$
  end
  $g^e \leftarrow$ keep the last model of the cycle
  if best model better than start **then**
    set best model as start model: $g^s \leftarrow g^*$
    reset cycle counter $\psi \leftarrow 0$
    $\mathcal{B} \leftarrow$ `MapAggregation`$(g^*, g^e, \mathcal{X})$
  else
    $\mathcal{B} \leftarrow$ `MapAggregation`$(0, g^e, \mathcal{X})$
    start a new cycle $\psi \leftarrow \psi + 1$
  end
  `Train Discriminator`$(\mathcal{D}_t \cup \mathcal{B})$
end

3. Lookahead Adversarial Learning (LoAd)

Robustness and divergence issues of adversarial networks are not secret to anyone (Arjovsky et al., 2017; Goodfellow et al., 2014b; Liu & Hsieh, 2019; Roth et al., 2017; Salimans et al., 2016) and we had to tackle that in our semantic segmentation setup. We take inspiration from "lookahead optimizer" (Zhang et al., 2019) and allow the adversarial network to *go ahead* and actually diverge (to some extent) helping us to gain new insights and construct new datasets of label maps from these divergent (or degraded) models. Inspired by the idea of DAGGER (Ross et al., 2011), we aggregate these new datasets in a buffer and use them for

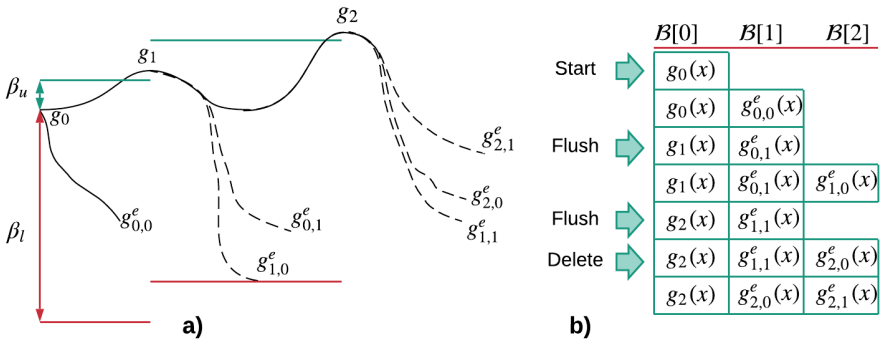

Figure 1. a) hypothetical convergence graph, b) corresponding label map aggregation buffer.

retraining the discriminator at the end of every "cycle" of LoAd. Next, we go back to where the divergence started (similar to "1 step back" in lookahead optimizer) to improve our next predictions and avoid further divergence. Note that these new datasets are not designed or generated adversarial examples but sequentially degraded label maps.

Algorithm 1 provides a pseudo code level description of LoAd. Let us assume a starting model $g^s = g_0$ (e.g., the end model after Stage 1 training as explained in Section 2). First, we evaluate the model on a subset of validation data (a hold-out set) to understand our current mean-intersection-over-union (mIoU, μ in Algorithm 1). This serves as both starting and current best mIoU (μ^s and μ^*, respectively). We can already train our discriminator for the first time using $\mathcal{D}_t \cup g_0(\mathcal{X}_t)$, a set composed of full training data (images and maps) plus a set of generated (fake) maps. With this, we have initialized our label map aggregation buffer with $\mathcal{B} = \mathcal{B}[0] = g_0(\mathcal{X}_t)$. We then continue training adversarial until one of the following two criteria is met: a) patience iteration counter γ reaches its maximum Γ, alerting us that it is enough looking ahead, b) we diverge (in mIoU sense) reaching a pre-defined lower-bound ($\mu^s - \beta_l$) w.r.t. to the starting mIoU μ^s. If any of the two criteria are met, the cycle is finished, and we pick the last model of the cycle denoted by g^e. Throughout each cycle, we also seek for an updated model offering a mIoU better than the staring one, and if such a new best peak model g^* (above an upper-bound $\mu^s + \beta_u$) is found, the cycle would be returning two models, the best model of the cycle g^* and its ending model g^e. Per cycle one or both of these models $\{g^*, g^e\}$ would be passed to our map aggregation algorithm to generate new "fake" label maps which will be aggregated in $\mathcal{B}$. This dynamically updated dataset in $\mathcal{B}$ concatenated with $\mathcal{D}_t$ will then be used to retrain the discriminator before the next cycle starts. At the end of each cycle, we go back and restart training adversarial from the newly found peak g^* or the old starting point g^s. Lastly, if we do more than Ψ cycles from a starting model g^s and a new peak is not found to replace it, the algorithm fully stops and returns the overall best model.

To make this crystal clear, we use a hypothetical convergence graph in Fig. 1 a) and corresponding dynamically updated map aggregation buffer depicted in Fig. 1 b) to walk you through what LoAd does in action. As can be seen in Fig. 1, starting from g_0, the first adversarial cycle immediately descends towards divergence ending with $g^e_{0,0}$. We denote the jth cycle spawned from the ith peak with $g_{i,j}$. Note that $g^e_{0,0}$ does not descend by β_l, and thus, we are assuming that the cycle is ended due to reaching patience limit of Γ propagations (or iterations) as described in Algorithm 1. This cycle also did not introduce a new peak better than g_0. Thus, only $g^e_{0,0}(\mathcal{X})$ will be added as a new set to the buffer $\mathcal{B}$. This is where we go back and restart adversarial training from g_0, but this time with a retrained discriminator. As can be seen in the figure, this helps to ascend towards g_1 after which we diverge again in the second cycle. So, the second cycle returns a new peak $g^* = g_1$ as well as the ending model $g^e = g^e_{0,1}$ for map aggregation. Since a new peak is found (better than g^0), we flush the buffer filling it with $\mathcal{B} = [g_1(\mathcal{X}) \,|\, g^e_{0,1}(\mathcal{X})]$ as shown in Fig. 1 b). Any cycle that only returns an ending model (an no new peak) would lead to creating a new label map set added to the end of the buffer unless the buffer is full; i.e., it already contains B_{max} label map sets. In that case, we first delete the label map set corresponding to the oldest end model and then the new label map set is added to the end of $\mathcal{B}$. An example of this scenario in our hypothetical setup is where the set corresponding to $g^e_{1,1}$ is deleted in favor of the newcomer set corresponding to $g^e_{2,1}$.

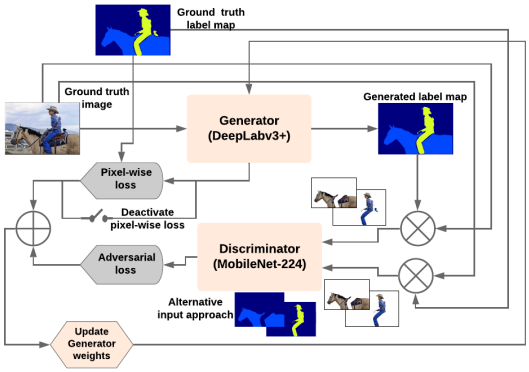

Figure 2. Adversarial network architecture.

Table 1. Performance comparison on PASCAL VOC 2012 *validation* set.

Method	backg.	aero.	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
DeepLabv3+	95.55	90.33	**44.23**	89.56	72.15	81.11	**96.76**	**91.37**	**94.33**	**51.87**	**96.08**
DLv3+ &LoAd	**95.65**	**92.92**	43.86	**90.31**	**77.73**	**82.02**	96.29	90.18	94.20	51.73	95.55
contd.	d.table	dog	horse	m.bike	person	p.plant	sheep	sofa	train	tv	mIoU
DeepLabv3+	60.14	**92.63**	93.33	**89.23**	90.18	67.19	**93.75**	61.26	**94.81**	80.27	82.20
DLv3+ &LoAd	**64.28**	92.14	93.30	88.63	90.19	**67.65**	93.65	**64.93**	94.18	80.31	**82.84**

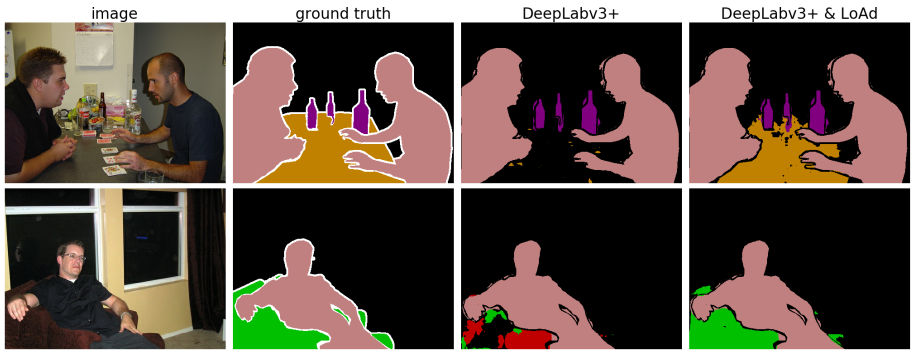

Figure 3. Selected qualitative results on PASCAL VOC 2012 *validation* set.

4. Experimental Setup

Network architecture. The experimental network architecture is shown on Fig. 2. As can be seen, we opted for DeepLabv3+ with a modified Xception-65 backbone (Chen et al., 2018), bearing in mind that DeepLabv3+ might not be the easiest model to simply plug into an adversarial settings. We chose Mobilenet-224 (Howard et al., 2017) as our discriminator. The figure shows that the discriminator training policy is conditional on the input image split into different classes (using ground truth and generated label maps) and stacked into the input channels of the discriminator. Our trainings are run separately on Nvidia P100 Tesla nodes each with 16 GB of memory. Notably, we are particularly interested in models that *run fast at inference* time for *near real-time* field applications. That is why we picked a DeepLabv3+ base model that offers speed (no multi-scaling, no CRFs) and performance at the same time.

Adversarial training. To train the discriminator, we used a batch size of 16, and set $\alpha = 1$ with a dropout rate of 0.01 (Howard et al., 2017). We used Adagrad with learning rate $lr_d = 0.01$. For adversarial training, we used a batch size of 5 due to the memory limitation of the GPU nodes available. The adversarial learning rate in this setting was set to $lr_a = 2.5e - 7$, and we trained using a momentum of 0.95. We used Gumbel softmax (Jang et al., 2016) with temperature $\tau = 20$ which in effect boosts the lr_a. Adversarial training is conducted based on LoAd (in Algorithm 1) with $\beta_u = 0.1\%$ in mIoU and $\beta_l = 5\%$ in mIoU. Patience counter maximum is set to $\Gamma = 50$, and maximum allowed cycles is set to $\Psi = 50$. Maximum buffer size is set to $B_{max} = 3$.

PASCAL VOC 2012 dataset. PASCAL VOC 2012 dataset (Everingham et al., 2015) contains 20 foreground object classes and 1 background class. It contains $1,464$ train, $1,449$ validation, and $1,456$ test pixel-level annotated images. For the experiments on this dataset, we started from DeepLabv3+ checkpoint pre-trained on PASCAL VOC 2012 achieving mIoU $= 82.2\%$ (see, Table 5 in (Chen et al., 2018)) followed by applying LoAd without any further fine-tuning. Since the images have a shape of at most 512×512 pixels, we used a crop size of 513×513 for Deeplabv3+ input layer following (Chen et al., 2018). For early stopping evaluation when training the discriminator as well as for evaluating the mIoU during adversarial training we used 30% (500 images) of the validation set (as hold-out set).

5. Evaluation Results

Table 1 summarizes the performance comparison between the baseline (DeepLabv3+, also abbreviated as DLv3+) and the proposed boosted model after applying LoAd (DeepLabv3+ & LoAd) on the full validation set. The results are interesting in the sense that even though the overall mIoU has increased by 0.6%, in some of the highlighted classes such as "aeroplane", "sofa", "diningtable", and "boat" the improvement in IoU ranges from $+2\%$ to $+5\%$ which is quite significant. Obviously, we degrade is some other classes mostly by a fraction of a percent, except for the "car" class where we degrade by more than 1%. Some qualitative results are illustrated in Fig 3. On the top row, the whole "dining table" is missed by the baseline and LoAd fully recovers that. The second row shows interesting signs of resolving class swap/confusion between "sofa" and "chair". Interested readers are referred to extensive quantitative and qualitative results in https://arxiv.org/abs/2006.11227.

References

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

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., and Chen, L.-C. Panoptic-deeplab. *arXiv preprint arXiv:1910.04751*, 2019.

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

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Henry, C., Azimi, S. M., and Merkle, N. Road segmentation in sar satellite images with deep fully convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(12):1867–1871, 2018.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y., and Yang, M.-H. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Ke, T.-W., Hwang, J.-J., Liu, Z., and Yu, S. X. Adaptive affinity fields for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 587–602, 2018.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9404–9413, 2019.

Lin, T., Jin, C., and Jordan, M. I. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*, 2019.

Liu, X. and Hsieh, C.-J. Rob-GAN: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11234–11243, 2019.

Liu, Z., Li, X., Luo, P., Loy, C. C., and Tang, X. Deep learning markov random field for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1814–1828, 2017.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3431–3440, 2015.

Luc, P., Couprie, C., Chintala, S., and Verbeek, J. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

Menze, M. and Geiger, A. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3061–3070, 2015.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Nouiehed, M., Sanjabi, M., Huang, T., Lee, J. D., and Razaviyayn, M. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, pp. 14905–14916, 2019.

Ostrovskii, D. M., Lowy, A., and Razaviyayn, M. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems. *arXiv preprint arXiv:2002.07919*, 2020.

Rezaei, M., Harmuth, K., Gierke, W., Kellermeier, T., Fischer, M., Yang, H., and Meinel, C. A conditional adversarial network for semantic segmentation of brain tumor. In *International MICCAI Brainlesion Workshop*, pp. 241–252. Springer, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pp. 2018–2028, 2017.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Shen, F., Gan, R., Yan, S., and Zeng, G. Semantic segmentation via structured patch prediction, context crf and guidance crf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1953–1961, 2017.

Shvets, A. A., Rakhlin, A., Kalinin, A. A., and Iglovikov, V. I. Automatic instrument segmentation in robot-assisted surgery using deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 624–628. IEEE, 2018.

Souly, N., Spampinato, C., and Shah, M. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5688–5696, 2017.

Volpi, M. and Ferrari, V. Semantic segmentation of urban scenes by learning local class interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1–9, 2015.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Xue, Y., Xu, T., Zhang, H., Long, L. R., and Huang, X. SegAN: Adversarial network with multi-scale L1 loss for medical image segmentation. *Neuroinformatics*, 16(3-4): 383–392, 2018.

Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., and Agrawal, A. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7151–7160, 2018a.

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pp. 9593–9604, 2019.

Zhang, Z., Zhang, X., Peng, C., Xue, X., and Sun, J. Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–284, 2018b.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2881–2890, 2017.

Zhao, S., Wang, Y., Yang, Z., and Cai, D. Region mutual information loss for semantic segmentation. In *Advances in Neural Information Processing Systems*, pp. 11115–11125, 2019a.

Zhao, S., Wu, B., Chu, W., Hu, Y., and Cai, D. Correlation maximized structural similarity loss for semantic segmentation. *arXiv preprint arXiv:1910.08711*, 2019b.