A IMPLEMENTATION DETAILS

Datasets. We evaluate our approach on the PASCAL VOC 2012 (Everingham et al., 2015) and the Cityscapes (Cordts et al., 2016) datasets. For PASCAL VOC, we follow Chen et al. (2017) to augment with the extra annotations provided by Hariharan et al. (2011). For Cityscapes, we follow the standard evaluation protocol in Cordts et al. (2016).

During the surrogate parameter search, we randomly sample 1500 training images in PASCAL VOC and 500 training images in Cityscapes to form the hold-out set $\mathcal{S}_{\text{hold-out}}$, respectively. The remaining training images form the training set $\mathcal{S}_{\text{train}}$ in search. After the search procedure, we re-train the segmentation networks with the searched losses on the full training set and evaluate them on the actual validation set.

Implementation Details. We use Deeplabv3+ (Chen et al., 2018) with ImageNet-pretrained ResNet-50/101 (He et al., 2016) backbone as the network model. The segmentation head is randomly initialized. In surrogate parameter search, the backbone is of ResNet-50. μ_0 is set to make $g(y;\theta) = y$. The training and validation images are down-sampled to be of 128×128 resolution. In SGD training, the mini-batch size is of 32 images, and the training is of 1000 iterations. The initial learning rate is 0.02, which is decayed by polynomial with power 0.9 and minimum learning rate 1e-4. The momentum and weight decay are set to 0.9 and 5e-4, respectively. For faster convergence, learning rate of the segmentation head is multiplied by 10. The search procedure is conducted for $T = 20$ steps, and $M = 32$ loss parameters are sampled in each step. In PPO2 (Schulman et al., 2017), the clipping threshold $\epsilon = 0.1$, and μ_{t+1} is updated by 100 steps. After surrogate parameter search, the re-training settings are the same as Deeplabv3+ (Chen et al., 2018), except that the loss function is substituted by the searched surrogate loss function. The backbone is of ResNet-101 by default.

B PARAMETERIZATION WITH PIECEWISE LINEAR FUNCTIONS

Here we choose the continuous piecewise linear function for parameterizing $g(y;\theta)$, where the form of constraints and parameters are very similar to that of the piecewise Bézier curve described in Section 4.2. Experimental results on PASCAL VOC 2012 (Everingham et al., 2015) are presented at the end of this section.

A continuous piecewise linear function consists of multiple line segments, where the right endpoint of one line segment coincides with the left endpoint of the next. Suppose there are n line segments in a piecewise linear function, then the k-th line segment is defined as

$$A(k,s) = (1-s)A_k + sA_{k+1}, \quad 0 \le s \le 1 \tag{15}$$

where s transverses the k-th line segment, $A_k = (A_{k,u}, A_{k,v})$ and $A_{k+1} = (A_{(k+1),u}, A_{(k+1),v})$ are the left endpoint and right endpoint of the k-th line segment, respectively, in which u, v index the 2-d plane axes.

To parameterize $g(y;\theta)$ via continuous piecewise linear functions, we assign

$$y = (1-s)A_{k,u} + sA_{(k+1),u}, \tag{16a}$$

$$g(y;\theta) = (1-s)A_{k,v} + sA_{(k+1),v}, \tag{16b}$$

$$\text{s.t. } A_{k,u} \le y \le A_{(k+1),u}, \tag{16c}$$

where θ is the collection of all control points. Given an input y, the segment index k and the transversal parameter s are derived from Eq. (16c) and Eq. (16a), respectively. Then $g(y;\theta)$ is assigned as Eq. (16b). Because $g(y;\theta)$ is a function defined on $y \in [0,1]$, we arrange the endpoints in the u-axis as,

$$A_{0,u} = 0,\ A_{n,u} = 1,\ A_{k,u} < A_{(k+1),u},\ 0 \le k \le n-1 \tag{17}$$

where the u-coordinate of the first and the last endpoints are at 0 and 1, respectively.

We enforce the two constraints introduced in Section 4.1 on the searching space through parameters θ. These two constraints can be formulated as

$$A_{0,v} = 0,\ A_{n,v} = 1; \qquad \textit{(truth-table constraint)}$$

$$A_{k,v} \le A_{k+1,v},\ k = 0, \ldots, n-1. \qquad \textit{(monotonicity constraint)}$$

In practice, we divide the domain $[0, 1]$ into n subintervals uniformly, and fix the u-coordinate of endpoints at the intersections of these intervals, i.e., $A_{k,u} = \frac{k}{n}$ where $0 \leq k \leq n$. Then the specific form of the parameters is given by

$$\theta = \{n(A_{k+1,v} - A_{k,v}) \mid k = 0, \ldots, n-1\}.$$

According to the constraints, the parameters need to satisfy

$$\frac{1}{n}\sum_{k=0}^{n-1} \theta_k = 1,\ \theta_k \geq 0, k = 0, \ldots, n-1.$$

In order to meet the above constraints, during the surrogate parameter search, we first sample parameters from a normal distribution without truncation, and then apply Softmax operation on the sampled parameters. The normalized parameters are used as the actual parameters for piecewise linear functions.

In our implementation, we use piecewise linear functions with five line segments. The effectiveness are presented in Table 7. The searched losses parameterized with piecewise linear functions are on par or better the previous losses on their target metrics, and achieve very similar performance with that of using piecewise Bézier curve for the parameterization.

Table 7: Performance of different metrics on PASCAL VOC. Piecewise linear functions are used for the parameterization in our method. The results of each loss function's target metrics are underlined. The scores whose difference with the highest is less than 0.3 are marked in **bold**.

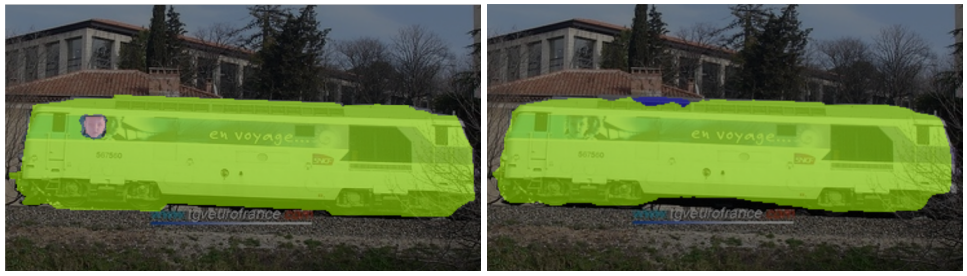
Loss Function	mIoU	FWIoU	BIoU	BF1	mAcc	gAcc
Cross Entropy	78.69	91.31	70.61	65.30	87.31	**95.17**
WCE (Ronneberger et al., 2015)	69.60	85.64	61.80	37.59	**92.61**	91.11
DPCE (Caliva et al., 2019)	79.82	**91.76**	71.87	66.54	87.76	**95.45**
SSIM (Qin et al., 2019)	79.26	91.68	71.54	66.35	87.87	**95.38**
DiceLoss (Milletari et al., 2016)	77.78	91.34	69.85	64.38	87.47	95.11
Lovàsz (Berman et al., 2018)	79.72	**91.78**	72.47	66.65	88.64	**95.42**
Searched mIoU	**80.94**	**92.01**	73.22	67.32	90.12	**95.46**
Searched FWIoU	79.05	**91.78**	71.47	64.24	89.77	**95.31**
Searched BIoU	43.62	70.50	**75.37**	46.23	53.21	82.60
Searched BF1	1.87	1.03	6.85	**76.02**	6.54	2.17
Searched mAcc	74.33	88.77	65.96	46.81	**92.34**	93.26
Searched gAcc	78.95	91.51	69.90	62.65	88.76	**95.19**
Searched mIoU + BIoU	**81.24**	**92.48**	**75.74**	68.19	90.03	**95.42**
Searched mIoU + BF1	79.11	91.38	71.71	73.55	89.28	**95.17**

C VISUALIZATION AND DISCUSSION ON BOUNDARY SEGMENTATION

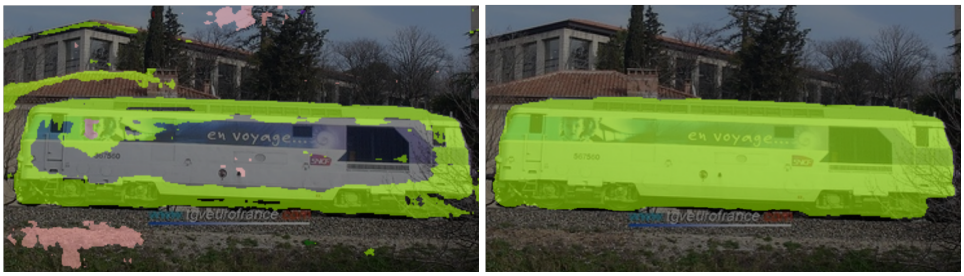
During the re-training stage, we find the segmentation result trained with surrogates for BIoU and BF1 metrics particularly interesting. To further investigate their properties, we visualize the segmentation results trained with surrogates for boundary metrics.

Boundary segmentation. As shown in Table 2 and Table 7, despite the great improvement achieved on BIoU and BF1 scores by training with surrogate losses for BIoU and BF1, respectively, other metrics show a significant drop. Fig. 4 and Fig. 5 visualizes the segmentation results of surrogate losses for mIoU, BIoU/BF1, and mIoU + BIoU/BF1. It can be seen that the surrogate losses for BIoU/BF1 guide the network to focus on object boundaries but ignore other regions, thus fail to meet the needs of other metrics. Training with surrogate losses for both mIoU and BIoU/BF1 can refine the boundary meanwhile maintain good performance for mIoU.

Boundary tolerance of the BF1 metric. Boundary metrics (e.g., BIoU and BF1) introduce the tolerance for boundary regions to allow slight misalignment in boundary prediction. Interestingly, we find that using the surrogate loss for BF1 with non-zero tolerance will lead to sawtooth around



(a) Ground-truth (b) Searched mIoU



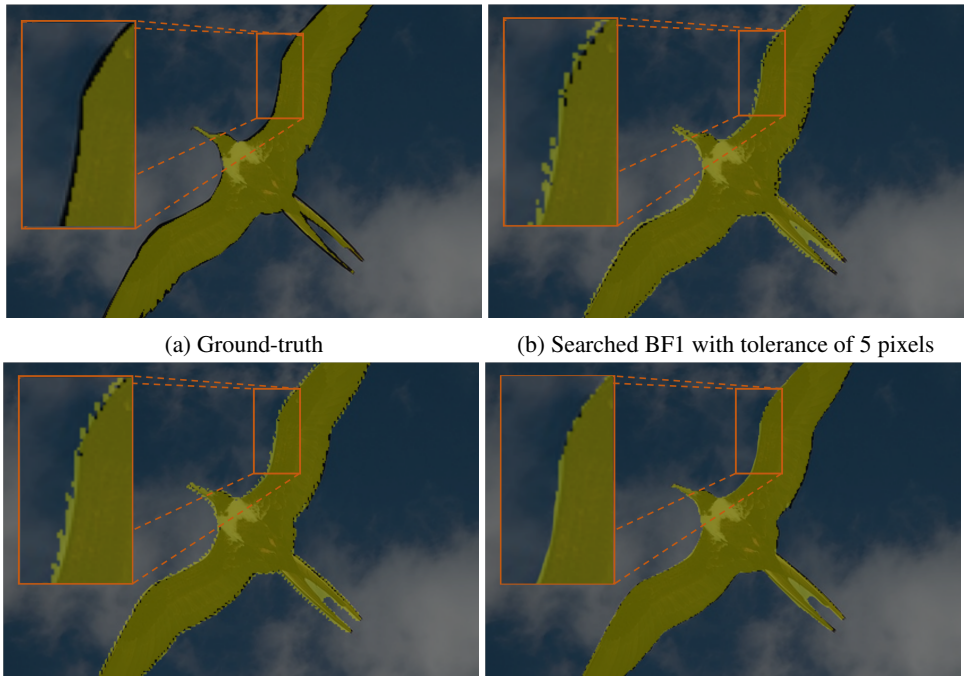
(c) Searched BIoU (d) Searched mIoU + BIoU

Figure 4: Segmentation results of surrogate losses for mIoU and BIoU.

(a) Ground-truth (b) Searched mIoU (c) Searched BF1 (d) Searched mIoU + BF1

Figure 5: Segmentation results of surrogate losses for mIoU and BF1.

the predicted boundaries, as shown in Fig. 6. Such sawtooth waves are within the tolerance range, which would not hurt the BF1 scores. When the boundary tolerance range in BF1 score reduces, the sawtooth phenomenon gets punished. The corresponding surrogate losses are learned to remove such sawtooth waves.



(a) Ground-truth (b) Searched BF1 with tolerance of 5 pixels

(c) Searched BF1 with tolerance of 2 pixels (d) Searched BF1 with tolerance of 0 pixel

Figure 6: Segmentation results of surrogate loss for mIoU + BF1, with different BF1 tolerance ranges.