Uncertainty Reduction for Model Adaptation in Semantic Segmentation

Appendix

Prabhu Teja S
Idiap Research Institute & EPFL
prabhu.teja@idiap.ch

François Fleuret
University of Geneva & Idiap Research Institute
francois.fleuret@unige.ch

A. Toy example's network architecture

For the toy experiment presented in Section 3.1, we use a network from Kristiadi et al. [38]. The architectural details of the network are given in Table 5.

Table 5: Architecture of the network used for the toy experiment in Figure 3

Layer name	**Description**
Feature Extractor	[Linear 2×20 BatchNorm ReLU Linear 20×2 ReLU]
Classifier	Softmax(2)

B. Entropy measurements

In Figure 1 and in Section 2.1, we glossed over the details of the toy-problem. We provide the details here.

Referring to Figure 1, we are trying to reason scenarios that, in the absence of labeled target data, can be expected to result in good target performance. Our argument is that reducing the uncertainty of predictions on the target domain is the an effective strategy to improve performance on the target domain. In order to do so, we concoct toy scenarios, and analyse their uncertainties. For this illustration, we use entropy as a measure of the uncertainty.

Let us consider a two-class classification problem as shown in Figure 1. Let X be the feature random variable, and $Y \in \{0, 1\}$ be the labels. Let us assume that the class conditional distributions be normally distributed *i.e.* $X|Y = k \sim \mathcal{N}(\mu_k, \sigma^2)$. Thus $\mu_X(x) = \frac{1}{2}\left(\mathcal{N}(x; \mu_0, \sigma^2) + \mathcal{N}(x; \mu_1, \sigma^2)\right)$, assuming uniform prior on Y. Let the threshold random variable T with a density $\mu_T(t)$. This distribution is determined by a learning algorithm. Let $\hat{Y}$ be the random variable denoting the predictions whose probability is computed using a sigmoid on the feature and a threshold as

$$\rho(x; t) \equiv p(\hat{Y} = 1 | X = x; T = t) = \frac{1}{1 + e^{-(x-t)}}. \tag{7}$$

The entropy of the above defined categorical distribution is given by

$$\mathbb{H}(\hat{Y}|X = x, T = t) = -\left(\rho(x; t) \log(\rho(x; t)) + (1 - \rho(x; t)) \log(1 - \rho(x; t))\right) \tag{8}$$

The entropy of a classification over the entire domain X can be computed as

$$\mathbb{H}(\hat{Y}|T = t) = \int -\left(\rho(x; t) \log(\rho(x; t)) + (1 - \rho(x; t)) \log(1 - \rho(x; t))\right) \mu_X(x) dx \tag{9}$$

The above defined marginal entropy is defined for a specific choice of the threshold t. To see the how the feature distribution itself influences the generalization, the entropy over all possible thresholds is computed. The *total entropy* over all the thresholds possible is computed by integrating over the entire range of t.

$$\mathbb{H}(\hat{Y}) = \int\int -\left(\rho(x;t)\log(\rho(x;t)) + (1-\rho(x;t))\log(1-\rho(x;t))\right)\mu_X(x)\mu_T(t)dxdt \tag{10}$$

We emphasize that t is the set of thresholds that can be generated by a learning algorithm. However, we simplify it to using the domain X for this discussion.

We would like to train networks that result in low overall entropy in Equation (10). This coincides with out argument that the features that can be separated by several choices of threshold are likelier to generalize better. To visualize this, we run a few simulations that compute the marginal and total entropy for various settings of feature distributions. In Figure 4, we show the data distribution $\mu_X(x)$ for various μ_0, μ_1 values. Variance σ^2 is fixed to 1. In orange, we show marginalized entropy (Equation (9)). We scale it by a constant for plotting purposes.

It is very apparent that the higher is $|\mu_1 - \mu_2|$, lower the overall entropy. Given that the two Gaussians are the class conditional distributions, the Bayes optimal decision boundary, that can be computed to be $\frac{\mu_1+\mu_2}{2}$, coincides with the point where Equation (9) is minimized, for all scenarios except for the one with high overlap in Figure 4(a). Extending it, the lower is the Bayes risk, better is the generalization. Thus the least entropy separation is also the best separator we can achieve. We

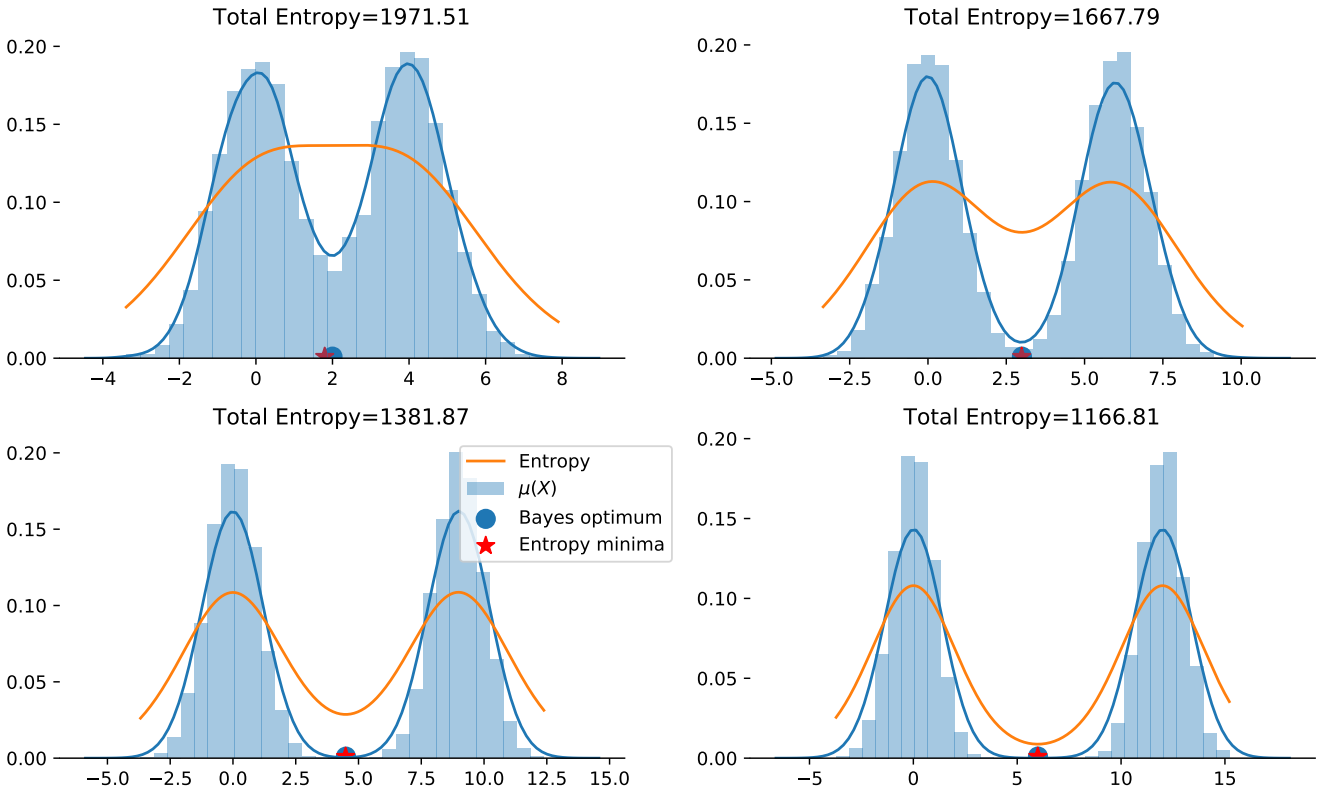

Figure 4: Illustration of the effect of the separation of the Gaussians on the entropy of the classification.

skip an important detail here: by extending the threshold search to a value that is on the extremities of the X-axis, we can get a threshold that results in an overall entropy that is very low. However, such a separator is practically useless. In our method, we avoid this by using the pseudo-labeling and the initializing the network with the source trained weights.

We can see that for this simple scenario that minimizing the entropy is a fairly good strategy in the absence of labels to find near optimal decision boundary.

C. Using squared error instead of entropy loss in Equation (2)

In Section 2.2, we proposed the uncertainty loss, and we use a squared error form instead of the standard entropy based uncertainty quantification. Here we provide a plausible explanation that follows the argument in Chen et al. [12]. Experimental evidence is provided in [54].

The traditional entropy regularizer has been used extensively in various applications like semi-supervised learning [29], domain adaptation [69]. However, the gradient of the entropy penalty with respect to the softmax output is not well behaved for probability is close to 1. We refer the reader to [12, Figure 1] for an illustration of this. However, squared loss and entropy loss can be viewed as a special case of f-divergence.

Let P and Q be two distributions with pdfs p and q their density functions. Then their f-divergence is defined as

$$D_f(P||Q) = \int_{\mathbb{R}} f\left(\frac{p(x)}{q(x)}\right) q(x)dx \tag{11}$$

We get KL-divergence by using $f(t) = t\log t$. Instead, using $f(t) = (t-1)^2$ gives the Max-squared loss formulation in [12]. They find that using this instead of $f(t) = t\log t$ results in better behavior for optimization. We find a similar empirical result that using the squared error form for Equation (2) results in better results than using posterior entropy.

D. Experimental ablations

D.1. Sensitivity to auxiliary decoders

In the numbers reported in Tables 1a and 1b, we use a dropout ratio of $p = 0.5$. We show an ablation on the dropout values in Table 6. Intuitively, this value indicates the level of noise resiliency that we expect in the network; a too low a value has very little use as it is equivalent to having a single decoder without dropout, and too high a value destroys too much information fed to the decoders. Thus an intermediate value like 0.5 is likely to be more appropriate for our use. We find that our experiments support this notion in Table 6. For this comparison, we use 5 auxiliary decoders. However, our proposed method is quite robust to this choice.

Dropout (p)	0.2	0.5	0.8
Performance (mIoU) %	44.44	45.07	44.14

Table 6: Optimal feature dropout proportions for GTA $\rightarrow$ Cityscapes experiment.

D.2. Number of auxiliary decoders

The auxiliary decoders play an important role in the level of feature stability induced. Indeed Gal and Ghahramani [25] show that larger the number of samples drawn from the dropout distribution, better is the approximation. In Table 7, we see that the number of decoders plays an important role, but the method is fairly stable in its performance over a range of decoder count.

# Decoders	1	3	4	5
Performance (mIoU)	43.5	44.20	44.67	45.07

Table 7: Performance variation to the number of auxiliary decoders for GTA $\rightarrow$ Cityscapes experiment.

D.3. Using single dropout decoder

In our method we propose the use of multiple decoders instead of one, and we show the effect of using a single decoder to decode the noisy features in Table 8. While it conceptually seems that using a single decoder should suffice, we see that using multiple auxiliary decoders is helpful. Using multiple decoders forces the feature extractor to be more robust, whereas using a single decoder forces the decoder to be more stable. However, the ASPP decoder is one layer deep, its representation capacity is limited and thus it is unable to do so, as evidenced in Table 8.

Method	mIoU %
Single decoder with dropout	43.29
Multiple auxiliary decoders	45.07

Table 8: Utility of using multiple decoders

E. Qualitative results for Cityscapes to Cross City Adaptation

In Figures 5 to 8 we show some qualitative improvements from our results for the *CS-CC* experiments.

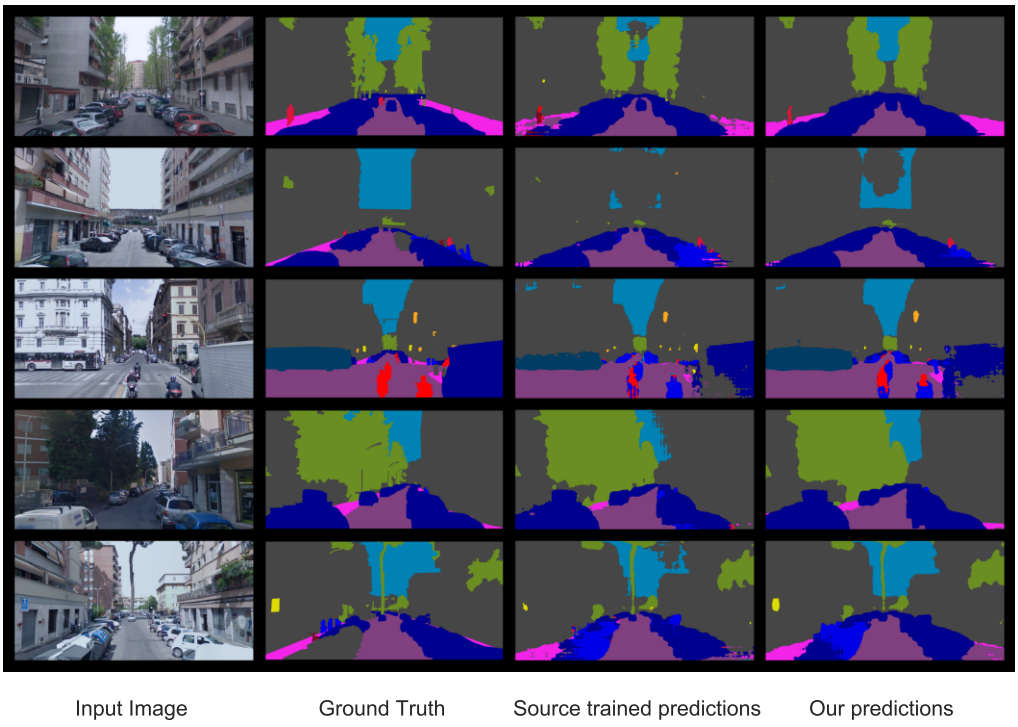

Figure 5: Best five case results of adaptation for the Cityscapes to Rome (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

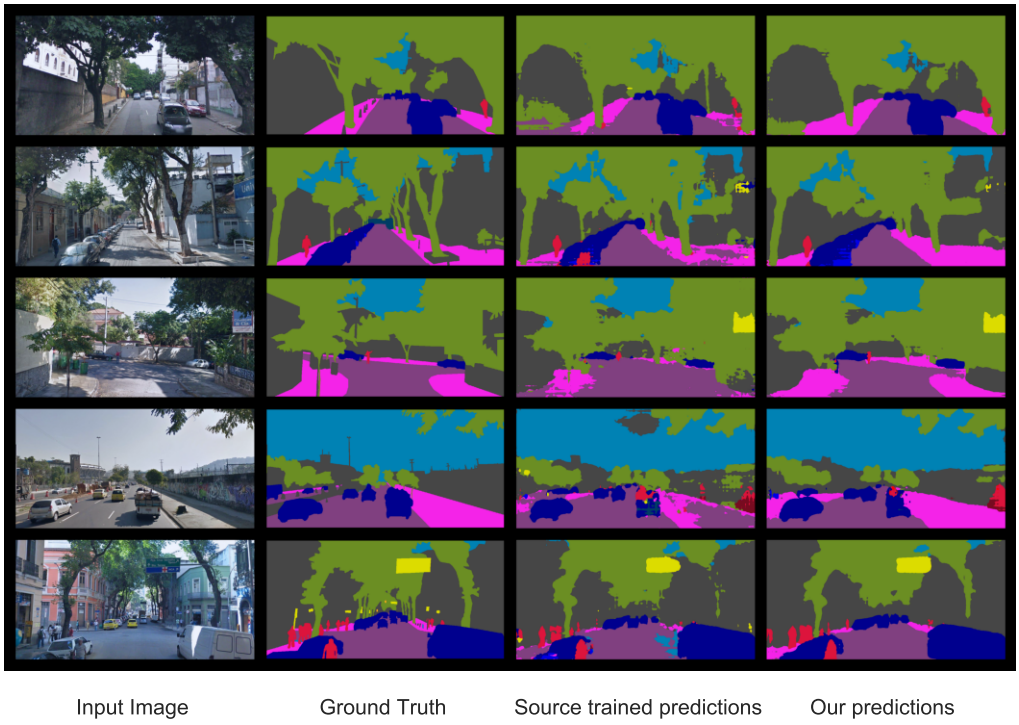

Figure 6: Best five results of adaptation for the Cityscapes to Rio (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

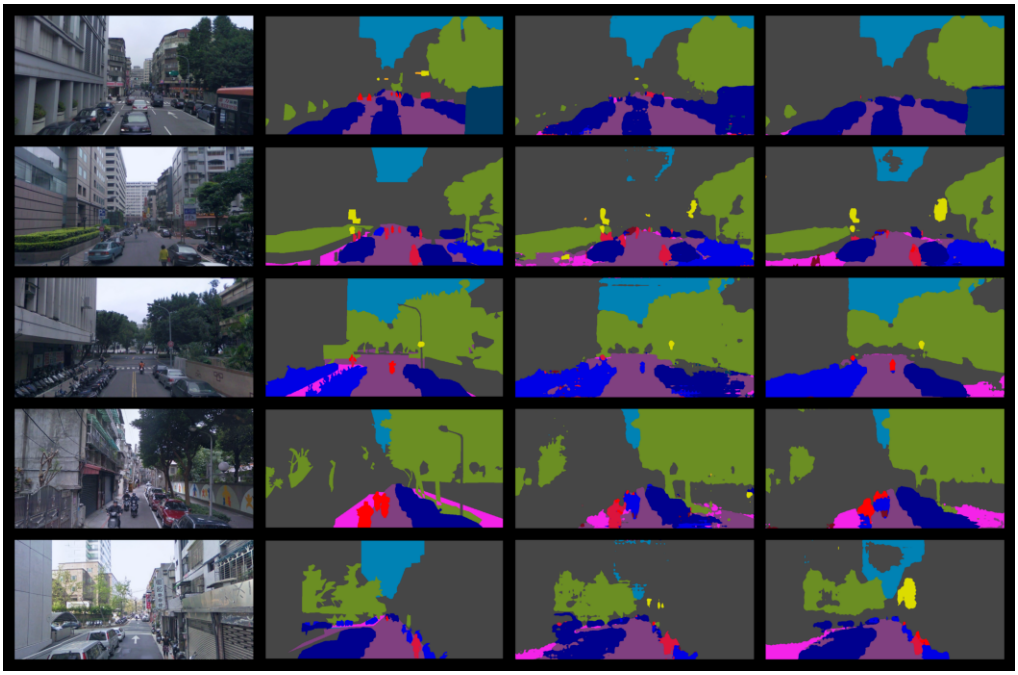

Figure 7: Best five results of adaptation for the Cityscapes to Taipei (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

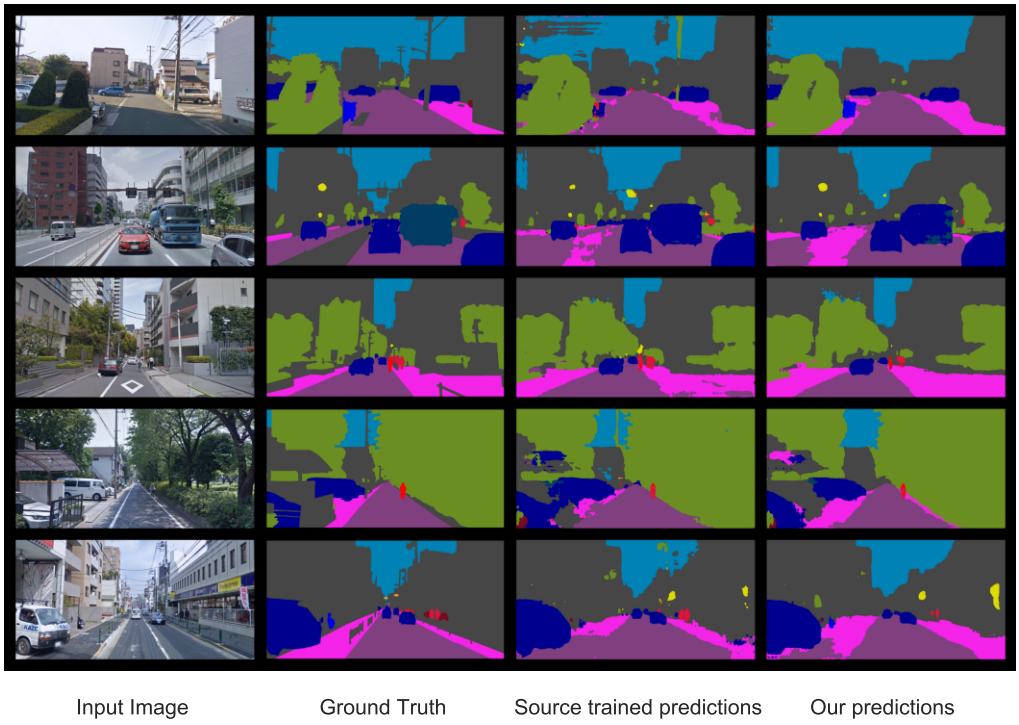

Figure 8: Best five results of adaptation for the Cityscapes to Tokyo (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

E.1. Failure cases

In Figures 9 to 12, we show the five images that have least improved over the adaptation process.

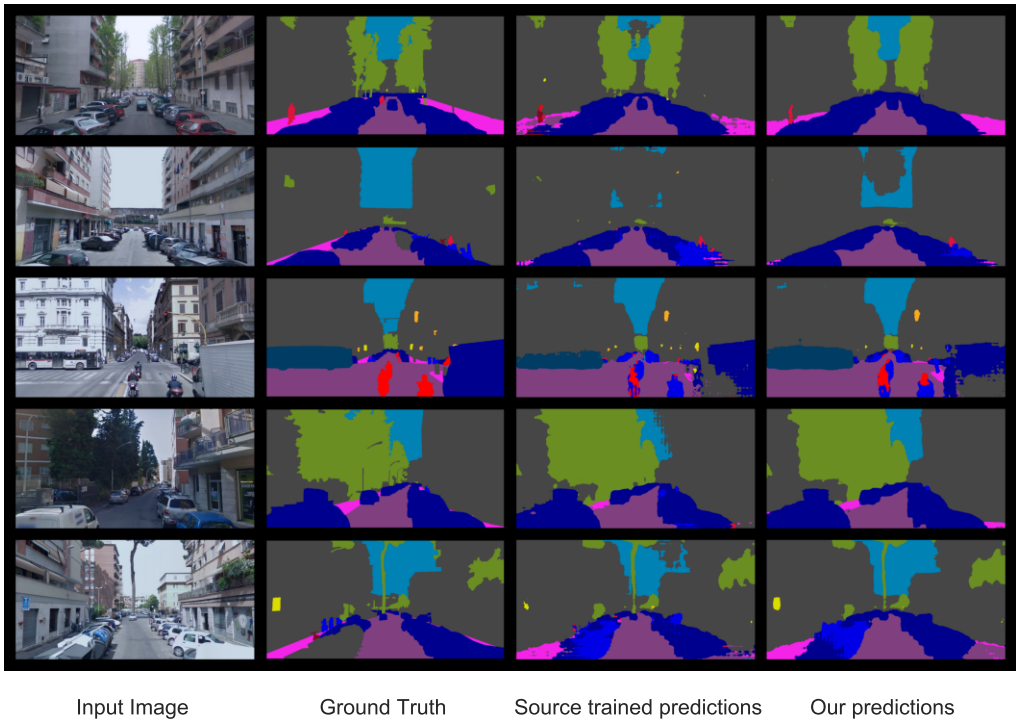

Figure 9: Worst five results of adaptation for the Cityscapes to Rome (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

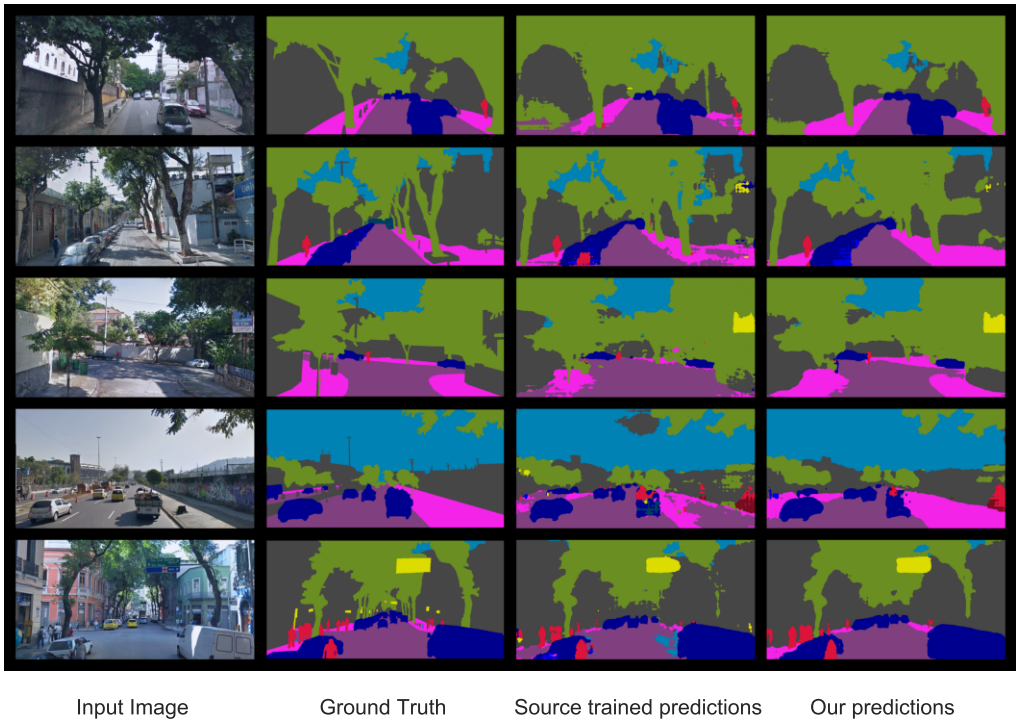

Figure 10: Worst five results adaptation for the Cityscapes to Rio (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

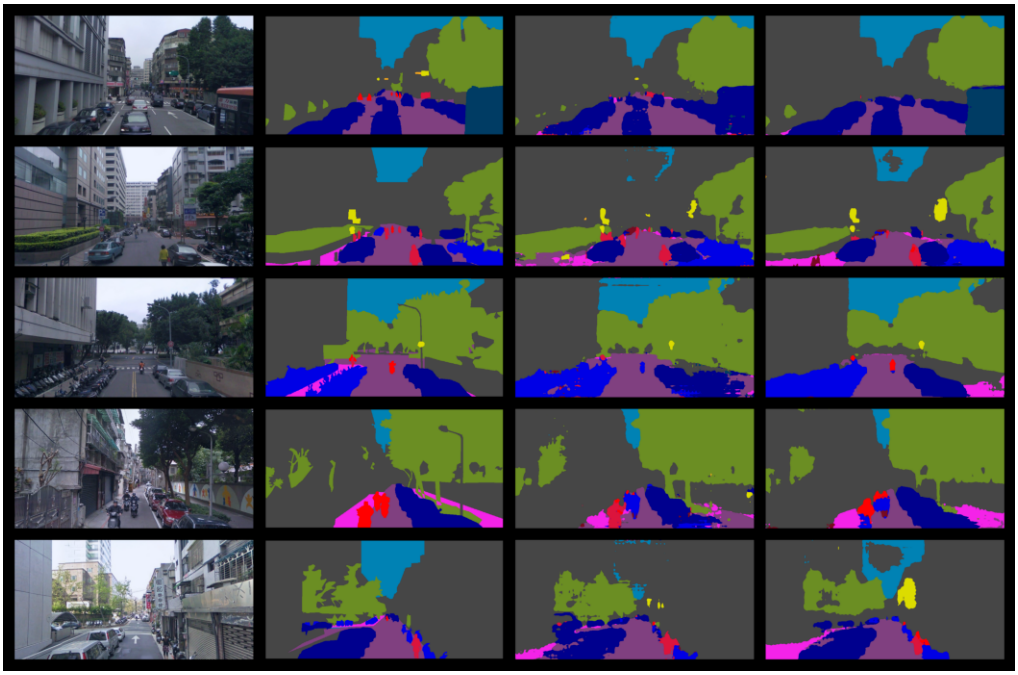

Figure 11: Worst five results adaptation for the Cityscapes to Taipei (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.

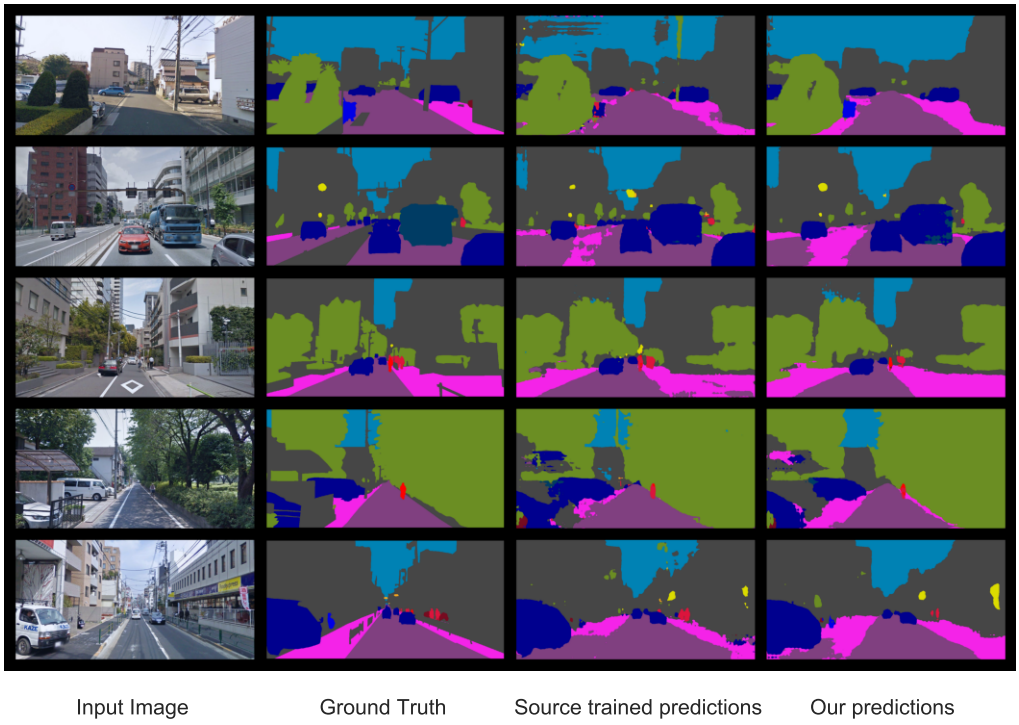

Figure 12: Worst five results adaptation for the Cityscapes to Tokyo (Cross City). The first column is the test image, the second column is the ground truth, the third and fourth columns are the source trained model, and results of our proposed method.