Spatial Consistency Enhanced Dissimilarity Coefficient based Weakly Supervised Object Detection - Appendix

Aditya Arun, C.V. Jawahar, M. Pawan Kumar

APPENDIX A
LEARNING OBJECTIVE

In this section, we provide a detailed derivation of the objective function presented in Section 4.2 of the paper.

Given the loss function Δ (equation (7) of the main paper), which is tuned for the task of object detection, we compute the diversity terms as given in equation (9) of the main paper. Recall that the diversity for any two distributions is the expected loss of the samples drawn from the two distributions. For the prediction distribution Pr_p and the conditional distribution Pr_c, we derive the diversity between them and their self diversities as follows.

Diversity between prediction net and conditional net: Following equation (9) of the main paper, the diversity between prediction and conditional distribution can be written as,

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c) = \mathbb{E}_{\mathbf{y}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{y}_c \sim \mathrm{Pr}_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\Delta(\mathbf{y}_p, \mathbf{y}_c)]]. \tag{1}$$

The task specific loss function is decomposed over the bounding boxes as given in equation (7) of the main paper. We then write the expectation with respect to the conditional distribution (the inner distribution) as expectation over the random variables $\mathbf{z}$ with distribution $\mathrm{Pr}(\mathbf{z})$ using the Law of the Unconscious Statistician (LOTUS).

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c) = \mathbb{E}_{\mathbf{y}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{z} \sim \mathrm{Pr}(\mathbf{z})}[\frac{1}{B}\sum_{i=1}^{B}\Delta(\mathbf{y}_p^{(i)}, \hat{\mathbf{y}}_c^{k,(i)})]]. \tag{2}$$

The expectation over the random variable $\mathbf{z}$ with distribution $\mathrm{Pr}(\mathbf{z})$ is approximated by taking K samples from $\mathrm{Pr}(\mathbf{z})$,

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c) = \mathbb{E}_{\mathbf{y}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{B}\sum_{i=1}^{B}\Delta(\mathbf{y}_p^{(i)}, \hat{\mathbf{y}}_c^{k,(i)})]. \tag{3}$$

We finally compute the expectation with respect to the prediction distribution as,

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c) = \frac{1}{BK}\sum_{i=1}^{B}\sum_{k=1}^{K}\sum_{\mathbf{y}_p^{(i)}}\mathrm{Pr}_p(\mathbf{y}_p^{(i)};\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p^{(i)}, \hat{\mathbf{y}}_c^{k,(i)}). \tag{4}$$

Aditya Arun and C.V. Jawahar are with the CVIT, IIIT Hyderabad.

Self diversity for conditional net: As above, using equation (9) of the main paper, we write the self diversity coefficient of the conditional distribution as

$$DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c) = \mathbb{E}_{\mathbf{y}_c \sim \mathrm{Pr}_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\mathbb{E}_{\mathbf{y}'_c \sim \mathrm{Pr}_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\Delta(\mathbf{y}_c, \mathbf{y}'_c)]]. \tag{5}$$

We now write the two expectations with respect to the conditional distribution as the expectation over the random variables $\mathbf{z}$ and $\mathbf{z}'$ respectively. The task specific loss function is decomposed over the bounding box as shown in equation (7) of the main paper. Therefore, we re-write the above equation as

$$DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c) = \mathbb{E}_{\mathbf{z} \sim \mathrm{Pr}(\mathbf{z})}[\mathbb{E}_{\mathbf{z}' \sim \mathrm{Pr}(\mathbf{z})}[\frac{1}{B}\sum_{i=1}^{B}\Delta(\hat{\mathbf{y}}_c^{k,(i)}, \hat{\mathbf{y}'}_c^{k,(i)})]]. \tag{6}$$

In order to approximate the expectation over the random variables $\mathbf{z}$ and $\mathbf{z}'$, we use K samples from the distribution $\mathrm{Pr}(\mathbf{z})$ as

$$DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{K-1}\sum_{\substack{k'=1,\\ k'\neq k}}^{K}\frac{1}{B}\sum_{i=1}^{B}\Delta(\hat{\mathbf{y}}_c^{k,(i)}, \hat{\mathbf{y}'}_c^{k',(i)}). \tag{7}$$

On re-arranging the above equation, we get

$$DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c) = \frac{1}{K(K-1)B}\sum_{\substack{k,k'=1\\ k'\neq k}}^{K}\sum_{i=1}^{B}\Delta(\hat{\mathbf{y}}_c^{k,(i)}, \hat{\mathbf{y}'}_c^{k',(i)}). \tag{8}$$

Self diversity for prediction net: Similar to the above two cases, using equation (9) of the main paper, we can write the self diversity of the prediction net as

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_p) = \mathbb{E}_{\mathbf{y}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{y'}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\Delta(\mathbf{y}_p, \mathbf{y'}_p)]]. \tag{9}$$

We then decompose the task specific loss function over the bounding boxes as described in equation (7) of the main paper,

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_p) = \mathbb{E}_{\mathbf{y}_p \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{y'}_1 \sim \mathrm{Pr}_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\frac{1}{B}\sum_{i=1}^{B}\Delta(\mathbf{y}_p^{(i)}, \mathbf{y'}_p^{(i)})]] \tag{10}$$

Note that the prediction distribution is a fully factorized distribution, and we can compute its exact expectation. Therefore, we compute the two expectations with respect to the prediction distribution as,

$$\mathbb{E}_{\mathbf{y}_p\sim \Pr_p(\mathbf{y}'|\mathbf{x};\boldsymbol{\theta}_p)}[\frac{1}{B}\sum_{i=1}^{B}\sum_{\mathbf{y'}_p^{(i)}} \Pr_p(\mathbf{y'}_p^{(i)};\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p^{(i)},\mathbf{y'}_p^{(i)})]$$
$$= \frac{1}{B}\sum_{i=1}^{B}\sum_{\mathbf{y}_p^{(i)}}\sum_{\mathbf{y'}_p^{(i)}} \Pr_p(\mathbf{y}_p^{(i)};\boldsymbol{\theta}_1)\Pr_p(\mathbf{y'}_p^{(i)};\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p^{(i)},\mathbf{y'}_p^{(i)}) \tag{11}$$

APPENDIX B
OPTIMIZATION

A. Optimization over Prediction Distribution

As parameters $\boldsymbol{\theta}_c$ of the conditional distribution are constant, the learning objective of the prediction distribution (equation (15) of the main paper) results in a fully supervised training of the Fast-RCNN network [1]. Note that the only difference between the training of a standard Fast-RCNN architecture and our prediction net is the use of the dissimilarity objective function (equation (15) of the main paper) instead of minimizing the multi-task loss of the Fast-RCNN.

The prediction net takes as the input an image and the K predictions sampled from the conditional net. Treating these predictions of the conditional net as the pseudo ground truth label, we compute the gradient of our dissimilarity coefficient based loss function. As the objective given in equation (15) of the main paper is differentiable with respect to parameters $\boldsymbol{\theta}_p$, we update the network by employing stochastic gradient descent.

B. Optimization over Conditional Distribution

A non-differentiable training procedure: The conditional net is modeled using a Discrete DISCO Net which employs a sampling step from the scoring function $\mathcal{S}^k(\mathbf{y}_c)$. This sampling step makes the objective function non-differentiable with respect to the parameters $\boldsymbol{\theta}_c$, even though the scoring function $\mathcal{S}^k(\mathbf{y}_c)$ itself is differentiable. However, as the prediction network is fixed, the above objective function reduces to the one used in Bouchacourt *et al.* [2] for fully supervised training. Therefore, similar to Bouchacourt *et al.* [2], we solve this problem by estimating the gradients of our objective function with the help of temperature parameter ϵ as,

$$\nabla_{\boldsymbol{\theta}_c} DISC^{\epsilon}_{\Delta}(\Pr_p(\boldsymbol{\theta}_p),\Pr_c(\boldsymbol{\theta}_c)) = \pm\lim_{\epsilon\to 0}\frac{1}{\epsilon}(DIV^{\epsilon}_{\Delta}(\Pr_p,\Pr_c) - \gamma DIV^{\epsilon}_{\Delta}(\Pr_c,\Pr_c)) \tag{12}$$

where,

$$DIV^{\epsilon}_{\Delta}(\Pr_p,\Pr_c) = \mathbb{E}_{\mathbf{y}_p\sim\Pr_p(\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{z}_k\sim\Pr(\mathbf{z})}[\nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_a) - \nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_c)]] \tag{13}$$

$$DIV^{\epsilon}_{\Delta}(\Pr_c,\Pr_c) = \mathbb{E}_{\mathbf{z}_k\sim\Pr(\mathbf{z})}[\mathbb{E}_{\mathbf{z'}_k\sim\Pr(\mathbf{z})}[\nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_b) - \nabla_{\boldsymbol{\theta}_c}\mathcal{S}^{k'}(\hat{\mathbf{y}}'_c)]] \tag{14}$$

and,

$$\begin{aligned}\hat{\mathbf{y}}_c &= \arg\max_{y\in\mathcal{Y}}\mathcal{S}^k(\mathbf{y}_c)\\ \hat{\mathbf{y}}'_c &= \arg\max_{y\in\mathcal{Y}}\mathcal{S}^{k'}(\mathbf{y}_c)\\ \hat{\mathbf{y}}_a &= \arg\max_{y\in\mathcal{Y}}\mathcal{S}^k(\mathbf{y}_c)\pm\epsilon\Delta(\mathbf{y}_p,\hat{\mathbf{y}}_c)\\ \hat{\mathbf{y}}_b &= \arg\max_{y\in\mathcal{Y}}\mathcal{S}^k(\mathbf{y}_c)\pm\epsilon\Delta(\hat{\mathbf{y}}_c,\hat{\mathbf{y}}'_c)\end{aligned} \tag{15}$$

In our experiments, we fix the temperature parameter ϵ as $\epsilon = +1$.

Intuition for the gradient computation: We now present an intuitive explanation of the computation of gradient, as given in equation (12). For an input $\mathbf{x}$ and two noise samples $\mathbf{z}_k, \mathbf{z}_{k'}$, the conditional net outputs two scores $\mathcal{S}^k(\mathbf{y}_c)$ and $\mathcal{S}^{k'}(\mathbf{y}_c)$, with the corresponding maximum scoring outputs $\hat{\mathbf{y}}_c$ and $\hat{\mathbf{y}}'_c$. The model parameters $\boldsymbol{\theta}_c$ are updated via gradient descent in the negative direction of $\nabla_{\boldsymbol{\theta}_c} DISC^{\epsilon}_{\Delta}(\Pr_p(\boldsymbol{\theta}_p),\Pr_c(\boldsymbol{\theta}_c))$.

- The term $DIV^{\epsilon}_{\Delta}(\Pr_p,\Pr_c)$ updates the model parameters towards the maximum scoring prediction $\hat{\mathbf{y}}_c$ of the score $\mathcal{S}^k(\mathbf{y}_c)$ while moving away from $\hat{\mathbf{y}}_a$, where $\hat{\mathbf{y}}_a$ is the sample corresponding to the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c)\pm\epsilon\Delta(\mathbf{y}_p,\hat{\mathbf{y}}_c)$ with respect to the fixed prediction distribution samples $\mathbf{y}_p$. This encourages the model to move away from the prediction, which provides high loss with respect to the pseudo ground truth labels.
- The term $\gamma DIV^{\epsilon}_{\Delta}(\Pr_c,\Pr_c)$ updates the model towards $\mathbf{y}_b$ and away from the $\hat{\mathbf{y}}_c$. Note the two negative signs giving the update in the positive direction. Here $\mathbf{y}_b$ is the sample corresponding to the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c)\pm\epsilon\Delta(\hat{\mathbf{y}}_c,\hat{\mathbf{y}}'_c)$ with respect to the other prediction $\hat{\mathbf{y}}'_c$, encouraging greater diversity between $\hat{\mathbf{y}}_c$ and $\hat{\mathbf{y}}'_c$.

Training algorithm for conditional net: Pseudo-code for training the conditional network for a single sample from weakly supervised data is presented in algorithm 1 below. In algorithm 1, statements 1 to 3 describe the sampling process and computing the loss augmented prediction. We first sample K different predictions $\hat{\mathbf{y}}^k_c$ corresponding to each noise vector $\mathbf{z}_k$ in statement 2. For the sampled prediction $\hat{\mathbf{y}}^k_c$ we compute the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c)\pm\epsilon\Delta(\mathbf{y}_p,\hat{\mathbf{y}}_c)$. This is then used to find the loss augmented prediction $\hat{\mathbf{y}}_a$ given in statement 3.

In order to compute the gradients of the self diversity of conditional distribution, we need to find the maximum loss augmented prediction $\mathbf{y}_b$. Here, the loss is computed between a pair of K different predictions of the conditional net that we have already obtained. This is shown by statements 4 to 7 in algorithm 1.

For the purpose of optimizing the conditional net using gradient descent, we need to find the gradients for the objective function of the conditional net defined in equation (16) of the main paper. The computation of the unbiased approximate gradients for the individual terms in the objective function is shown in statement 8. We finally optimize the conditional net by the employing gradient descent step and updating the

model parameters by descending to the approximated gradients as shown in statement 9 of algorithm 1.

Algorithm 1: *Conditional net training algorithm*

Input : Training input $(\mathbf{x}, \mathbf{a}) \in \mathcal{W}$, and prediction net output $\mathbf{y}_p$

Output: $\hat{\mathbf{y}}_c^1, \ldots, \hat{\mathbf{y}}_c^K$, sample K predictions from the model

1 **for** $k = 1 \ldots K$ **do**

2 Sample noise vector $\mathbf{z}_k$, generate output $\hat{\mathbf{y}}_c^k$:

$$\hat{\mathbf{y}}_c^k = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c)$$

3 Find loss augmented prediction $\hat{\mathbf{y}}_a^k$ w.r.t. output from prediction net $\mathbf{y}_p$:

$$\hat{\mathbf{y}}_a^k = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\mathbf{y}_p, \hat{\mathbf{y}}_c^k)$$

4 Compute loss augmented predictions:

5 **for** $k = 1, \ldots, K$ **do**

6 **for** $k' = 1, \ldots, K, k' \neq k$ **do**

7 Find loss augmented prediction $\hat{\mathbf{y}}_b^k$ w.r.t. other conditional net outputs $\hat{\mathbf{y}}_c^k$:

$$\hat{\mathbf{y}}_b^{k,k'} = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\hat{\mathbf{y}}_c^k, \hat{\mathbf{y}}_c')$$

8 Compute unbiased approximate gradients for $DIV_\Delta^\epsilon(\mathrm{Pr}_c, \mathrm{Pr}_c)$ and $DIV_\Delta^\epsilon(\mathrm{Pr}_c, \mathrm{Pr}_c)$ as:

$$DIV_\Delta^\epsilon(\mathrm{Pr}_p, \mathrm{Pr}_c) = \frac{1}{KB} \sum_{k=1}^{K} \sum_{i=1}^{B} \Big[\nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_a^{(i)}) - \nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_c^{(i)}) \Big] \tag{16}$$

$$DIV_\Delta^\epsilon(\mathrm{Pr}_c, \mathrm{Pr}_c) = \frac{2}{K(K-1)B} \sum_{\substack{k,k'=1 \\ k' \neq k}}^{K} \sum_{i=1}^{B} \Big[\nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_b^{(i)}) - \nabla_{\boldsymbol{\theta}_c} \mathcal{S}^{k'}(\hat{\mathbf{y}}_c^{(i)'}) \Big] \tag{17}$$

Update model parameters by descending to the approximated gradients:

$$\boldsymbol{\theta}_c^{t+1} = \boldsymbol{\theta}_c^t - \eta \nabla_{\boldsymbol{\theta}_c} DISC_\Delta(\mathrm{Pr}_p(\boldsymbol{\theta}_p), \mathrm{Pr}_c(\boldsymbol{\theta}_c))$$

REFERENCES

[1] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[2] D. Bouchacourt, "Task-oriented learning of structured probability distributions," Ph.D. dissertation, University of Oxford, 2017.