Supplementary Material
Simultaneous Edge Alignment and Learning

Zhiding Yu[1], Weiyang Liu[3], Yang Zou[2], Chen Feng[4], Srikumar Ramalingam[5],
B. V. K. Vijaya Kumar[2], and Jan Kautz[1]

[1] NVIDIA {zhidingy, jkautz}@nvidia.com
[2] Carnegie Mellon University {yzou2@andrew, kumar@ece}.cmu.edu
[3] Georgia Institute of Technology wyliu@gatech.edu
[4] New York University cfeng@nyu.edu
[5] University of Utah srikumar@cs.utah.edu

1 Additional details

In this section, we present the additional details on evaluation benchmark and experiments which are not fully covered by the main paper. We believe these details will benefit successful reproduction of the proposed method and reported experiments. In addition, the rest of the supplementary material will also report more details regarding SBD re-annotation, as well as additional results on SBD and Cityscapes dataset.

1.1 Network hyperparameters

We use the code from [5], and exactly follow [5] to set the network hyperparameters, including learning rate, gamma, momentum, decay, etc. As a result, the learning rate and gamma on SBD/Cityscapes are respectively set as $1.0 \times 10^{-7}/5.0 \times 10^{-8}$ and 0.1/0.2. We keep the crop size as 472×472 on C-ityscapes, and unify the SBD crop size also as 472×472. For any method involving supervision with the unweighted sigmoid cross-entropy loss, we unify the learning rates on SBD/Cityscapes as 5.0×10^{-8} and 2.5×10^{-8}, while keeping other parameters the same as counter parts with reweighted loss.

1.2 SBD data split

In this work, we further randomly sample 1000 images from the original SBD training set as a **validation set**, while treating the rest 7498 images as a new training set. In addition, the original SBD test set with 2857 images remain as a held out **test set**. For the rest of this material, we will refer to the new training set with 7495 images as "**training set**", and the original SBD training set with 8498 images as "**trainval set**" for clarity.

To conduct parameter analysis and ablation study for the proposed framework, models with different parameters and modules are trained on the training set, and validated on the validation set. In addition, results reported on the SBD test set, including those reported in main paper Section 6.2 and the rest of this material, correspond to models trained on the trainval set.

1.3 Data augmentation and training label generation

We follow the preprocessing code of [5] to perform multi-scale data augmentation and generate slightly thicker edge labels for model training on SBD. In particular, we slightly modify the code to preserve instance-sensitive edges and augment both the SBD training and trainval set in Sec. 1.2, while keeping other implementation the same. We apply similar training label generation procedure to Cityscapes except removing the data augmentation.

Note that for evaluation under the "Raw" setting, raw predictions are matched with unthinned ground truths whose edge width is set to be the same as training edge labels. Under the "Thin" setting, on the other hand, the evaluation ground truth consists of single pixel wide edge labels.

1.4 Network training iteration numbers

Following [5], we report the performance of all models on SBD at 22000 iterations. For all models trained on cityscapes, the iteration number is empirically selected as 28000.

1.5 Computation cost

Taking SBD as example: On 1 TitanXP GPU + 2 Xeon E5-2640v4 CPUs, each net learning iter takes 7 seconds for iter_size of 10, while the alignment of every 300 images takes about 180 seconds using all 40 CPU threads in parallel, giving over 6000 training iters/day. For Cityscapes, the same platform can train about 4000 iters/day. We will add this info to the paper.

1.6 Benchmark parameters

In both [2] and [3], an important benchmark parameter is the matching distance tolerance which is the maximum slop allowed for correct matches of edges to ground truth during evaluation. The distance tolerance is often measured as proportion to the image diagonal. On the BSDS dataset [3,1], this parameter is by default set as 0.0075, while on SBD [2], the parameter is increased to 0.02 to compensate the increased annotation noise.

In light of these previous works, we follow [2] to set the matching distance tolerance as 0.02 for evaluations using the original SBD annotations. We also decrease the tolerance to 0.0075 for evaluations using the re-annotated high quality SBD test labels. In addition, given the high label quality and the large image diagonals, we decrease the tolerance in Cityscapes experiments to 0.0035. This corresponds to tolerating 8 pixels approximately on Cityscapes images. Note that our adopted Cityscapes benchmark is significantly stricter than [5] which followed [2] by setting the distance tolerance as 0.02 in their Cityscapes experiments.

Unlike cityscapes, SBD edge labels do not guarantee unified image border conditions. For some images, imperfect alignment between segmentation annotations and image borders leads to extra edges along image borders. As a result, we ignore edge evaluation within 5 pixels to image borders for all SBD experiments. For cityscapes, no border pixels are ignored since there is no such issue.

1.7 Color coding protocol

For experiments on both SBD and Cityscapes dataset, we follow the original color codings of PASCAL VOC and Cityscapes to visualize the semantic classes. In particular, the color of each pixel is visualized based on the following equation:

$$\mathbf{I} = \begin{cases} \mathbf{255} - (\sup_c P_c)\dfrac{\sum_{c=1}^{C} P_c(\mathbf{255} - \mathbf{M}_c)}{\sum_{c=1}^{C} P_c}, & \text{if } \sum_{c=1}^{C} P_c > 0 \\ \mathbf{255} & , \text{ Otherwise} \end{cases} \tag{1}$$

where c indicates the class index, and C the number of classes. P_c is the predicted edge probability of class c, while $\mathbf{M}_c$ is the RGB vector of class c following the dataset color coding. In addition, $\mathbf{255} \triangleq [255, 255, 255]^\top$.

2 SBD re-annotation

Although the SBD dataset provided instance-level segmentation labels with generally good qualities, we observe that a considerable portion of the labels exhibit the issue of having large misalignment and missing objects. This raises some concerns on the reliability of the evaluations based on the original labels, since the large distance tolerance may potentially compromise the accuracy on localization and precision-recall measurement. In light of this issue, we use LabelMe [4] to generate a high quality subset of the SBD test set with 1059 images. Fig. 1 illustrates an example of the re-annotation interface.

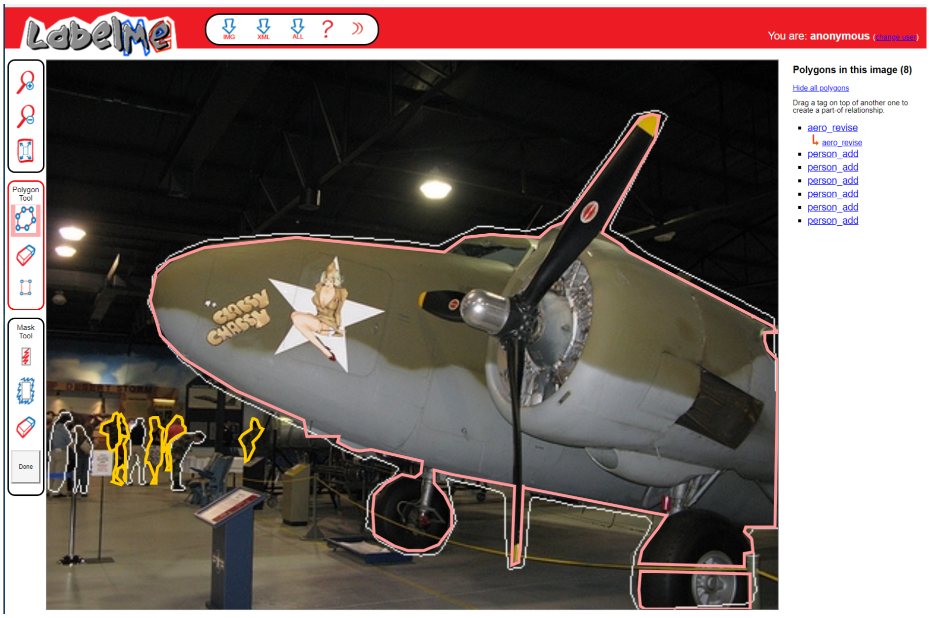
We also include all the re-annotated labels in this supplementary material in the form of Matlab files. Some examples of the re-annotated edge labels versus the original labels are also shown in Fig. 2 and Fig. 3.

3 Additional results on SBD

In this section, we report additional results on SBD which are not covered by the main paper.

3.1 Parameter analysis and ablation study

The main paper mentioned using the SBD validation set to determine σ_y and λ. Here, we comprehensively report the results corresponding to different parameters. Since we assume that our system does not have any knowledge on high quality annotations, we choose to validate parameters of the proposed framework under the original noisy SBD labels. Table 1 comprehensively reports the results corresponding to different parameter configurations of σ_x, σ_y, and λ.



You are: **anonymous** (change user)

Polygons in this image (8)

Hide all polygons

Drag a tag on top of another one to create a part-of relationship.

- aero_revise
  - aero_revise
- person_add
- person_add
- person_add
- person_add
- person_add
- person_add

Fig. 1. Illustration of SBD re-annotation with MIT LabelMe toolkit. In the image, white lines indicate the original SBD annotation, and colored polygons indicate re-annotated labels. One may notice the significant misalignment along the aeroplane boundary, as well as multiple persons with missing labels.

Table 1. Evaluation on SBD val set and 0.02 tolerance. Results are measured by maximum F-Measure (MF) at optimal dataset scale (ODS), measured by %.

Metric	$\sigma_x, \sigma_y, \lambda$	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MF (Thin)	4, 4, 0	85.1	73.8	80.7	59.1	68.2	82.3	79.9	81.7	54.6	76.4	46.8	80.4	82.4	74.7	80.4	52.6	75.6	48.3	74.0	65.2	71.1
	1, 4, 0	86.4	76.1	82.8	61.2	68.6	84.3	80.6	83.2	55.3	77.8	46.6	81.4	83.1	76.1	81.1	56.2	77.1	49.4	76.5	66.7	72.5
	1, 4, 0.02	86.0	77.3	82.6	60.6	68.7	84.0	81.8	83.9	56.7	76.5	48.1	81.9	84.3	77.6	81.8	58.2	76.0	49.8	78.3	69.0	**73.2**
	1, 4, 0.04	85.8	76.7	82.5	61.5	69.3	84.1	81.6	84.6	55.3	76.3	47.4	81.6	83.3	77.4	81.6	58.2	76.4	50.2	77.6	68.0	73.0
	1, 3, 0.04	85.7	77.3	83.3	59.8	69.5	84.6	81.8	84.2	55.9	77.1	47.8	82.1	83.8	77.2	81.5	57.9	77.7	49.9	78.2	67.3	73.1
	1, 2, 0.04	85.6	76.8	83.0	60.5	68.9	84.4	81.6	84.3	55.5	76.7	48.4	82.3	83.7	76.9	81.6	59.0	76.6	50.6	77.1	67.6	73.1

Note that we report results evaluated under the "Thin" setting with 0.02 tolerance. $\lambda = 0$ essentially means removing the Markov smoothness term in edge prior, while having $\sigma_x = \sigma_y = 4$ indicates removing the kernel bias. One could see that the above results indicate that both terms benefit edge learning and lead to better edge detector performance.

To better reveal the behavior of SEAL, we also include a comprehensive ablation study on the SBD test set with re-annotated labels in Table 2. We study with re-annotated labels since their high quality can capture the algorithm performance with best accuracy. Note that this experiment is purely for ablation study and is independent from parameter selection. Results show that alignment without smoothing produces the sharpest edges, but smoothing gives better

Table 2. Evaluation on SBD test set with re-annotated labels and 0.0075 tolerance. Results are measured by ODS-MF, with scores measured by %

Metric	$\sigma_x, \sigma_y, \lambda$	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MF (Thin)	$4, 4, 0$	75.99	61.89	72.56	49.11	63.62	74.31	66.54	74.21	48.09	68.62	38.63	74.89	72.18	62.71	75.00	48.25	72.06	49.61	67.89	53.66	63.49
	$1, 4, 0$	77.56	64.15	75.25	51.40	65.10	76.76	67.90	76.26	49.59	70.47	39.41	76.53	75.00	64.31	76.82	49.80	72.55	49.91	70.51	54.82	65.20
	$1, 4, 0.01$	77.68	65.27	76.04	51.98	68.45	79.43	70.44	78.68	49.96	70.96	40.48	77.74	74.91	65.92	78.27	48.86	73.72	51.33	73.51	57.15	66.54
	$\mathbf{1, 4, 0.02}$	77.64	65.70	76.07	52.08	68.39	79.83	70.64	79.06	49.76	71.05	40.66	78.23	75.35	66.08	78.21	49.39	74.06	51.12	73.46	57.36	**66.71**
	$1, 4, 0.04$	78.12	64.74	76.40	51.65	68.09	80.11	70.63	78.34	49.52	71.03	41.17	77.57	75.38	65.69	78.31	48.45	73.38	51.04	73.45	57.23	66.51
	$1, 3, 0.02$	77.52	64.77	76.03	52.48	67.83	79.50	71.05	78.62	49.57	71.89	40.66	77.76	75.27	65.60	78.37	47.76	73.07	51.16	73.87	57.65	66.52
	$1, 2, 0.02$	76.65	64.01	75.92	52.27	68.08	80.08	71.13	78.83	49.44	72.11	40.45	77.21	75.12	65.60	78.09	47.13	73.30	50.58	73.66	57.44	66.36
MF (Raw)	$4, 4, 0$	78.21	65.09	76.03	52.94	64.13	76.55	69.50	76.72	51.31	70.36	40.32	76.83	75.20	65.33	76.75	51.67	73.98	49.63	71.35	56.19	65.90
	$\mathbf{1, 4, 0}$	78.67	65.75	77.53	53.24	65.45	77.44	69.88	77.77	51.32	71.02	40.48	77.69	76.60	66.17	77.59	51.83	73.81	49.33	72.32	56.17	**66.50**
	$1, 4, 0.01$	75.54	59.83	75.81	50.72	65.81	75.92	68.41	75.51	49.98	68.60	39.57	74.64	72.90	62.94	74.37	47.94	72.44	48.55	70.42	56.48	64.32
	$1, 4, 0.02$	74.81	60.22	75.21	50.71	65.49	76.20	68.06	75.51	49.24	67.89	39.06	74.34	72.97	62.15	74.10	48.08	72.44	48.77	69.83	57.27	64.12
	$1, 4, 0.04$	75.27	59.76	75.53	50.27	65.15	76.35	68.11	74.93	49.09	67.55	39.63	73.96	72.88	61.46	73.95	48.39	71.73	48.21	70.01	56.52	63.94
	$1, 3, 0.02$	73.81	58.96	74.74	50.00	65.59	75.35	68.00	74.93	48.96	67.97	38.59	73.92	72.23	61.58	73.84	46.36	70.87	48.64	69.75	56.37	63.52
	$1, 2, 0.02$	72.24	57.27	73.44	49.09	64.29	74.50	67.33	73.28	48.15	67.63	38.15	71.91	71.30	60.43	72.47	45.52	70.84	47.62	68.32	55.69	62.47

trade-off. Decreasing σ_y (less alignment flexibility) drops performance in both settings.

3.2 Ground truth refinement

An important feature not covered in the main paper is that the ability to automatically refine noisy labels using the proposed framework. We conduct this experiment by running SEAL on the complete SBD dataset, and output the aligned edge labels upon convergence. Fig. 2 and Fig. 3 illustrate visualized results of the comparing methods.

One thing we observe is that dense-CRF tends to smooth out thinned structures such as human/chair legs, while these delicate structures are quite important in edge learning. In addition, dense-CRF sometimes also introduces noisy boundaries because of the limited representation power of low-level features. This partly explains why dense CRF overall does not even match the quality of the original labels. In fact, experiments on the SBD test set also indicate decreased model performance using dense CRF preprocessed labels.

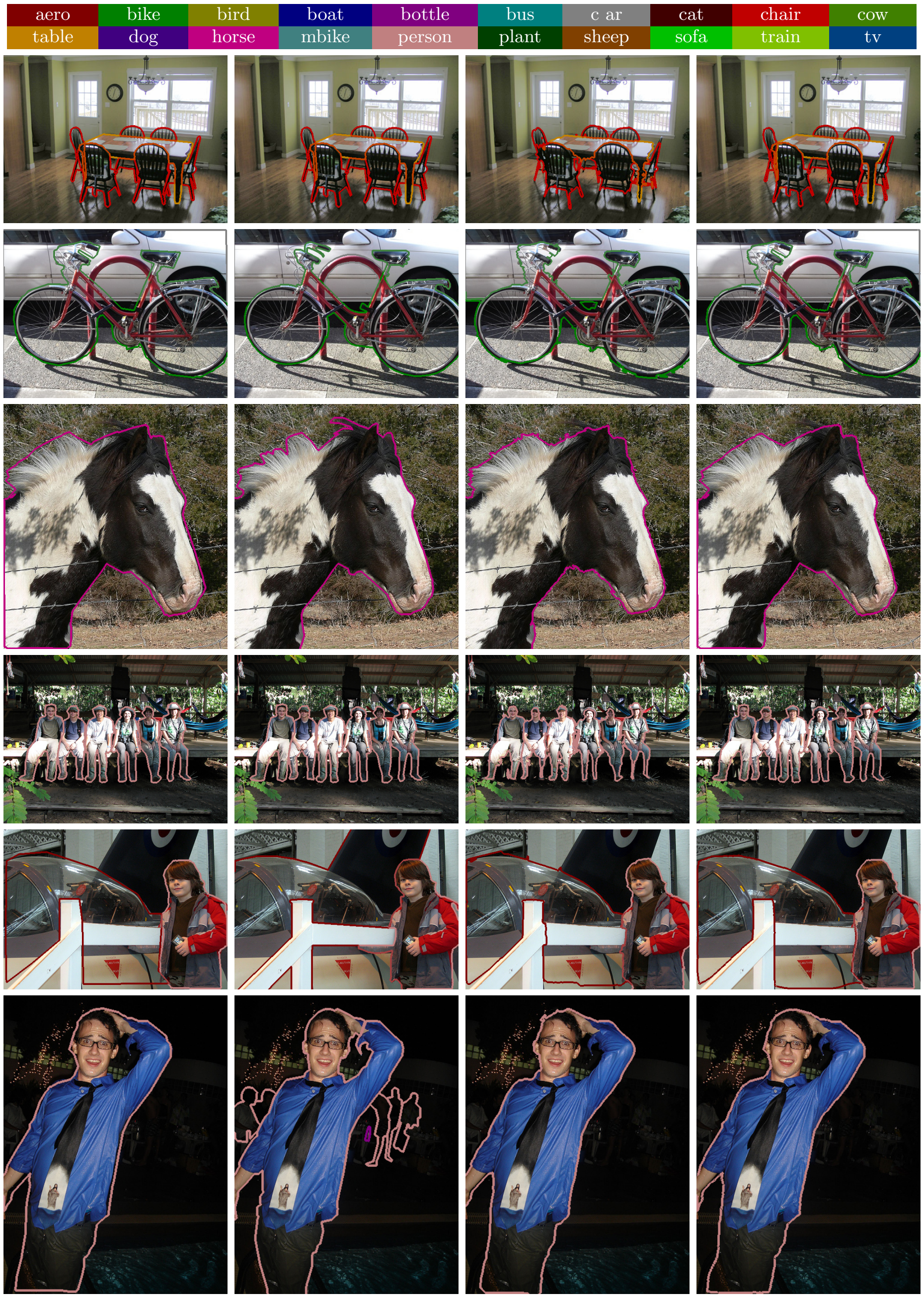


Fig. 2. Examples of annotations and aligned edge labels learned by different methods on SBD test set images. From left to right: original ground truth, re-annotated high quality ground truth, edge labels aligned via dense CRF, edge labels aligned via SEAL.

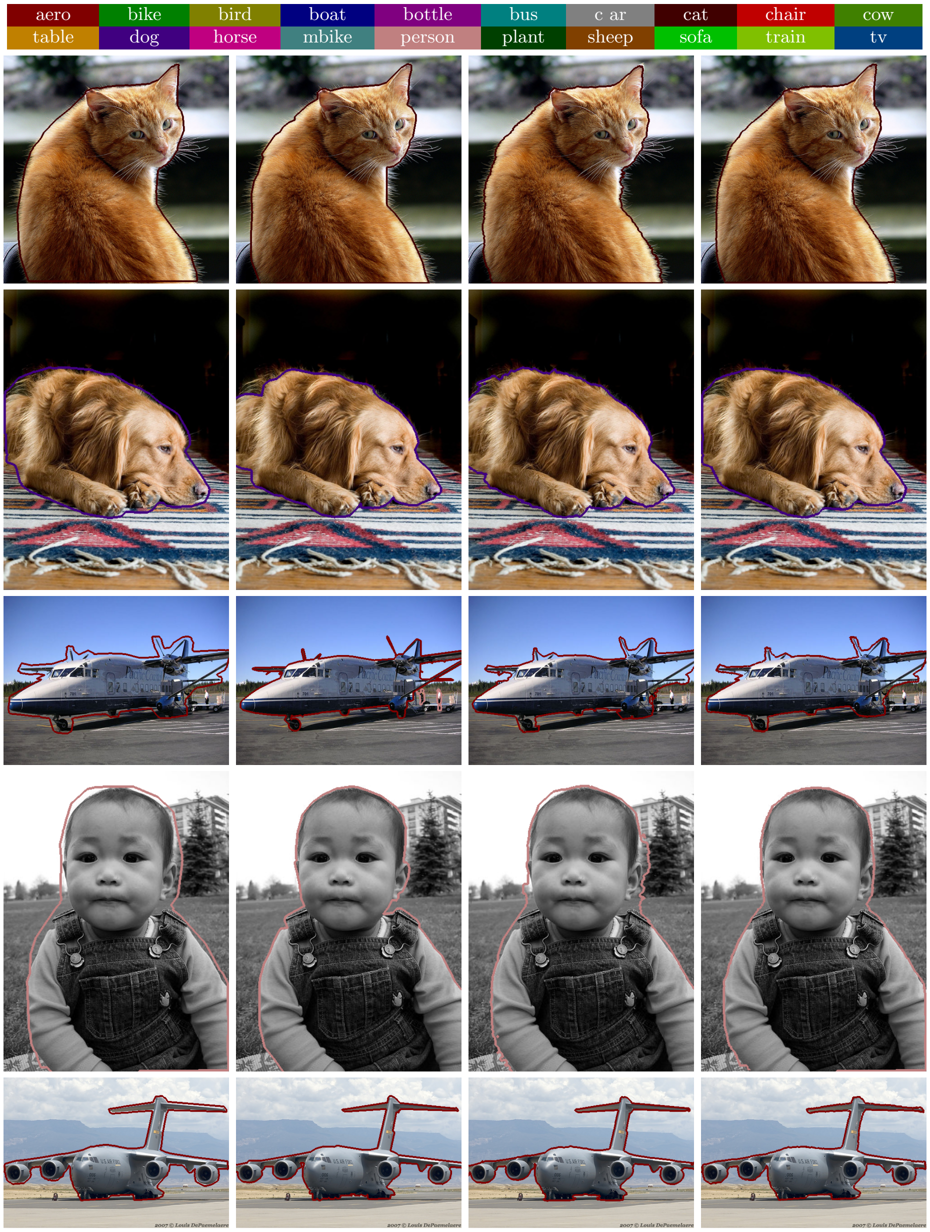


Fig. 3. Examples of annotations and aligned edge labels learned by different methods on SBD test set images. From left to right: original ground truth, re-annotated high quality ground truth, edge labels aligned via dense CRF, edge labels aligned via SEAL.

References

1. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. PAMI **33**(5), 898–916 (2011) 2
2. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011) 2
3. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. PAMI **26**(5), 530–549 (2004) 2
4. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV **77**(1-3), 157–173 (2008) 3
5. Yu, Z., Feng, C., Liu, M.Y., Ramalingam, S.: Casenet: Deep category-aware semantic edge detection. In: CVPR (2017) 1, 2