

A APPENDIX

A.1 PROOF OF LEMMA 2

Proof.

$$d_C(D^Y, D'^Y) = \sum_{i \in [1, k]} d_{JS}(D_i^Y, D_i'^Y),$$

According to the definition, the samples within the condition subsets share some label. So that, according to the previous definition of $d_{JS}(D, D')$, we have for every $i \in [1, k]$:

$$d_{JS}(D_i^Y, D_i'^Y) = 0,$$

□

A.2 PROOF OF LEMMA 3

Proof. According to Definition 3, we have

$$D_i^c \cap D_j'^c = \emptyset, i \neq j.$$

Therefore,

$$d_{JS}(D_i^c, D') = d_{JS}(D_i^c, D_i'^c).$$

□

A.3 PROOF OF CONDITIONAL ALIGNMENT

Proof. The situation of the large marginal difference on label space can be represented as follows. Given condition set $\mathcal{Y} = \{c_1, c_2, c_3, \dots, c_k\}$, for any $i \in [1, k]$, we have:

$$Y_i^c \cap Y'^c = Y_j'^c, i \neq j.$$

For convenience, we replace D^Y by Y .

Without loss of generality, we suppose $j = i + 1$, so that we have:

$$d_{JS}(Y, Y') = \sum_{i=1}^k d_{JS}(Y_i^c, Y_j'^c).$$

Specially, we set $Y_{k+1}'^c = Y_1'^c$.

We have conditional aligned domains D and D' , which can be represented as:

$$d_C(D, D') = 0.$$

Therefore, for any $i \in [1, k]$:

$$d_{JS}(D_i^c, D_i'^c) = 0.$$

We have conditional aligned D^Y and D'^Y , so it can instantly have:

$$d_{JS}(D_i'^c, Y_i'^c) = 0.$$

Combineing the equations above, we have:

$$d_{JS}(D_i^c, Y_i'^c) = 0.$$

According to Lemma 2, we have:

$$\begin{aligned} d_{JS}(Y_i^c, Y_j'^c) &= d_{JS}(D_i^c, Y_j^c) \\ &= d_{JS}(D_i^c, D_j^c) \\ &= d_{JS}(D_i^c, D_j'^c). \end{aligned}$$

It is possible to find an order of sorting the D_i^c and $D_i'^c$, so that the JS-convergence between D and D' can be:

$$d_{JS}(D, D') = \sum_{i=1}^k d_{JS}(D_i^c, D_j'^c).$$

Specifically, we set $D_{k+1}'^c = D_1'^c$. To this end, combining the equations above, we have:

$$d_{JS}(D, D') = d_{JS}(Y, Y').$$

□

A.4 DATASET DETAILS

In this section, we will provide details about the dataset we implemented in our experiments, including cell counting datasets and crowd counting datasets. Example visualizaion is showns as Figure 3.

For the crowd-counting task, the datasets include GTA5 Crowd Counting (GCC) (Wang et al., 2019b), UCF-QNRF (UCF) (Idrees et al., 2018), ShanghaiTech (SHA & SHB) (Zhang et al., 2016), and JHU-Crowd++Sindagi et al. (2022). The details of the crowd dataset are shown as follows:

- GCC (Wang et al., 2019b) is generated from multiple crowd scenes in Grand Theft Auto V, a video game, with 15,210 samples. The image size is 1920×1080 pixels. The synthetic environment contains multiple times of the day, seven types of weather, and diverse scenes (e.g. beach, street, and other common public scenes.). It provides various simulations of real-world scenes. The average of crowded count for each image is 500, with the highest count of 4000 and lowest count of zero.
- UCF (Idrees et al., 2018) is a large-scale dataset that contains 1535 high solution images with considerable crowd variation. The images are obtained from the Web by multiple platforms. So, the resolutions are highly dynamic. The average density of images is 1000 counts but with a standard deviation 7605.14.
- The ShanghaiTech (Zhang et al., 2016) dataset consists of parts A and B, containing 482 and 716 samples, respectively. Part A (SHA) is obtained from the Web with dynamic resolutions. The mean of counts per image is 541, with a standard deviation of 504. Part B (SHB) is retrieved from the security monitoring cameras on busy streets with fixed resolutions. The mean of counts per image is 122, with a standard deviation 93.
- The JHUCrowd++ (Sindagi et al., 2022) dataset consists of 4,372 images with detailed annotations, totaling approximately 1.51 million instances. The images are collected from diverse sources, including the web and surveillance cameras, featuring varying resolutions and perspectives. The dataset captures a wide range of crowd densities, from sparse to extremely dense scenes. The mean count per image is approximately 346, with a standard deviation of 1,094, indicating significant variability in crowd counts across the dataset.

The environments of the crowd datasets, including various weathers and scenes, are among the most challenging issues to handle in crowd counting. It requires algorithms with higher adaptability to handle it. Overall, the selection of datasets covers a sufficient variety of environments and scenes. In the following experiments, we examine the transferability of the BiAN by evaluating its performance in transferring features between the domains from the datasets shown above.

For the cell counting task, the datasets include three public benchmarks: synthetic fluorescence microscopy (VGG) dataset (Xie et al., 2018b), human subcutaneous adipose tissue (ADI) dataset (Cohen et al., 2017), and Dublin Cell Counting (DCC) dataset. The details of the cell dataset are shown as follows:

- VGG (Xie et al., 2018b) is a synthetic microscopy cell image dataset with 200 samples. It simulates bacterial cells from fluorescence-light microscopy at various focal distances. The size of microscopy images is maintained as 256×256 pixels. The cell amount of VGG for each image is 174 ± 64 .
- DCC (Marsden et al., 2018) dataset is built with 177 samples from various categories of cells from real cases, including embryonic mice stem cells, human lung adenocarcinoma, and monocytes. The image size ranges from 306×322 pixels to 798×788 pixels, due to obtained via dynamic zoom scope. Moreover, the cell amount for each image is 34 ± 21 , intended to increase the variation of the dataset.
- ADI (Cohen et al., 2017) is constructed from Genotype Tissue Expression Consortium (Lonsdale et al., 2013) with densely packed adipocyte cells from real cases. The dataset is built from 200 images. The image size is 150×150 pixels. The cell amount for each image is 165 ± 44 .

The slight deviation of the cell amount of each image provides a relative consistency in cell density. Various types of cells further challenge the performance of the model in the adaptability of scene presentation.

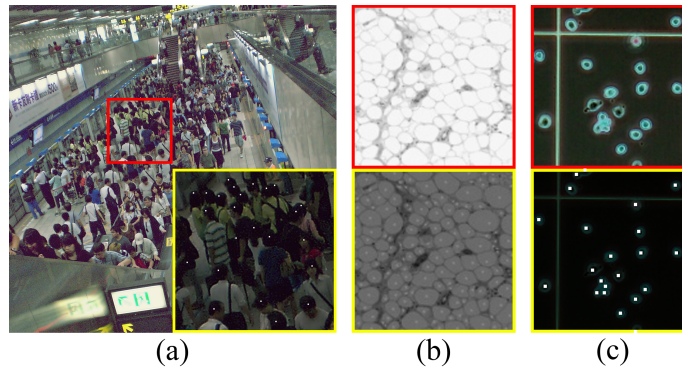


Figure 3: Object counting scenarios: (a) public security monitoring; (b) medical pathological analysis; (c) biological experiment.

A.5 EXPERIMENT IMPLEMENTATION DETAILS

We choose the Adam optimizer with decoupled weight decay. The learning rate for the optimizer is set to $1e-6$, and the weight decay rate is $1e-4$. For the learning rate, we use a step learning rate scheduler with a 10-epoch step to lower the learning rate by 0.1 for every step. To handle the limitation of GPU memory, we resize cell images to 128×128 pixels and crowd images to 96×96 pixels. Notably adopting the annotated counts before resizing the images to maintain the ground truth unaffected by squeezing. The coefficient α of CM loss is set to 100. Moreover, we apply the training scalar on the annotations to enhance the numeric difference. The scalar for VGG and ADI is 100. For DCC and all applied crowd datasets, it is set as 500, respectively. BiAN is fully implemented in PyTorch, running on a single NVIDIA RTX 3090 with a single Intel® Core™ i7-10700 CPU @ 2.90GHz.

A.6 ADDITIONAL EXPERIMENT ANALYSIS

A.6.1 SOURCE ON SYNTHETIC CROWD DATASET

Migrating from the source synthetic dataset to real-world dataset is a practical approach to handle insufficient data annotation issue. To validate BiAN performance on such condition, we conduct the experiments with the setting of GCC (*source*) and UCF (*target*). Specifically, we have to resize the input as smaller size (128×128 px) due to memory limitation. We have taken reasonable measure to preserve the information. The results still presents BiAN outperforms SOTA methods. But due to no guarantee on lost information, the experiment results only can be quantitatively referred.

Table 5: Counting MAE and MSE on crowd counting task from synthetic source. The best are highlighted in bold. DA: Domain Adaptation for short.

Methods	DA	GCC \rightarrow UCF	
		MAE \downarrow	MSE \downarrow
KDMG (Wan et al., 2022)	\times	99.5	173.0
UOT (Ma et al., 2021)	\times	83.3	142.3
STEERER (Han et al., 2023)	\times	74.3	128.3
Cycle GAN (Zhu et al., 2017)	\checkmark	257.3	400.6
SE CycleGAN (Wang et al., 2019b)	\checkmark	230.4	384.5
BiAN (Ours)	\checkmark	22.7	28.4

A.6.2 RELEVANCE ANALYSIS BETWEEN CONSISTENCY AND COUNTING RESULTS

In this section, we further demonstrate the proposed Condition-Consistency Mechanism (CM), which benefits from reliable counting when there is a lack of precise annotation during the adaptation process. We plot the curves presenting the tendency of MAE on the validation set and uncertainty during the training period. Specifically, the uncertainty index is calculated by the normalized CM loss $NORM(\mathcal{L}_{CM})$, indicating how inconsistent between features of assembling conditions and entire ones. It can be observed that the counting performance, which is inversely proportional to the MAE value, is promoted when the uncertainty index decays. Combined with the results in experiment results in Section 4.3, it can validate that the assumption on disjoint condition subsets is necessary in BiAN and conditional alignment framework.

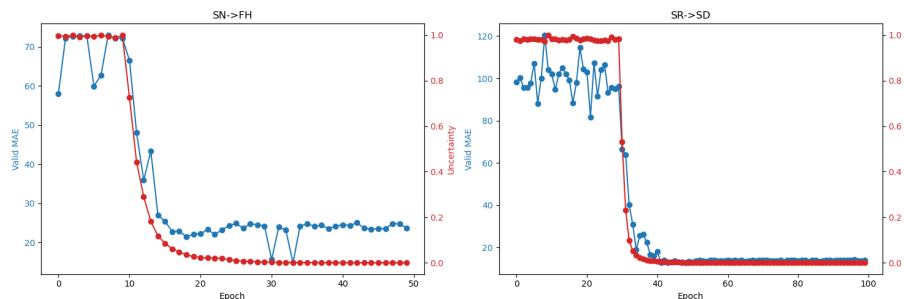


Figure 4: The tendency of validation counting MAE and the consistency on two domain combinations.

A.7 MODEL ARCHITECTURE OF BIAN

In this section, we present the architectural details of our proposed BiAN. As shown in Figure 2, the architecture can be divided into domain-specific feature extractors ($G_{S/T}$), density map regression layers (F), domain discriminator (C), condition-consistent layers (F_c). The total parameters amount of BiAN is 70,755,271 with an estimated model size of 2783.41 MB.

Table 6: Architecture of feature extractor in BiAN.

Layer (Type: Depth-Idx)	Output Shape	Parameters (#)
Conv2d: 3-1	[32, 256, 256]	384
Conv2d: 3-2	[32, 256, 256]	9,312
MaxPool2d: 3-3	[32, 128, 128]	–
Conv2d: 3-4	[64, 128, 128]	18,624
Conv2d: 3-5	[64, 128, 128]	37,056
MaxPool2d: 3-6	[64, 64, 64]	–
Conv2d: 3-7	[128, 64, 64]	74,112
Conv2d: 3-8	[128, 64, 64]	147,840
MaxPool2d: 3-9	[128, 32, 32]	–
Conv2d: 3-10	[256, 32, 32]	295,680
Conv2d: 3-11	[256, 32, 32]	590,592
SelfAttention: 3-12	[256, 32, 32]	263,424

The domain-specific feature extractors ($G_{S/T}$), detailed in Appendix A.7, are responsible for capturing relevant features from the input data in both source and target domains. Specifically, the input is resized as 256×256 px. And the network arguments are independent among G_S and G_T , but same architecture for similar feature retrieval.

The density map regression layers (F), detailed in Table 7, are designed to predict density maps from the extracted features. Specifically, it expands the channels of features by deconv operations and estimates the density. The output size is input-alike but only one channel, which is shown as resized 256×256 px. And the F_c shares weight and architecture with F for preparing partial results map for CM validation.

Table 7: Architecture of regression layers in BiAN.

Layer (Type: Depth-Idx)	Output Shape	Parameters (#)
Deconv2d: 3-13	[128, 64, 64]	131,456
Conv2d: 3-14	[128, 64, 64]	295,296
Conv2d: 3-15	[128, 64, 64]	147,840
Deconv2d: 3-16	[64, 128, 128]	32,960
Conv2d: 3-17	[64, 128, 128]	73,920
Conv2d: 3-18	[64, 128, 128]	37,056
Deconv2d: 3-19	[32, 256, 256]	8,288
Conv2d: 3-20	[32, 256, 256]	18,528
Conv2d: 3-21	[32, 256, 256]	9,312
Conv2d: 3-22	[1, 256, 256]	33

The domain discriminator (C), as described in Table 8, aims to align the domain distribution shift of features by broadcasting inverse gradient. The output is the binary label.

Table 8: Architecture of domain discriminate layers in BiAN.

Layer (Type: Depth-Idx)	Output Shape	Parameters (#)
Linear2d: 3-23	[256]	67,109,632
Linear2d: 3-24	[64]	16,576
Dropout1d: 3-25	[64]	–
Linear2d: 3-26	[2]	134
Softmax: 3-27	[2]	–

A.8 VISUALIZATION

In this section, we present the visual results of BiAN in the counting task experiments. As shown in Figure 6, we randomly select two samples from every cross-domain adaptation. In the visualization figure, we mark the inaccurate counts in the samples. The low-density samples can be counted in precise amounts, and the localization is also accurate. However, in microscopy cell images, cells of an overlapped or abnormal size are not fully recognized. The cell-alike objects (e.g. bubbles) easily distract the model recognition, especially in the DCC cell images. The conditional alignment mechanism enables BiAN to recognize distinguishing features of cells. As for the crowd counting task, human main characters are important cues to lead the model to marks. In contrast, the characters of hidden persons are easily missed targets. The results show that BiAN is able to retrieve the partial features of humans. It results in significant performance improvements. Overall, the visualization demonstrates the proposed model’s recognition ability and learning of the visual representation of counting targets.

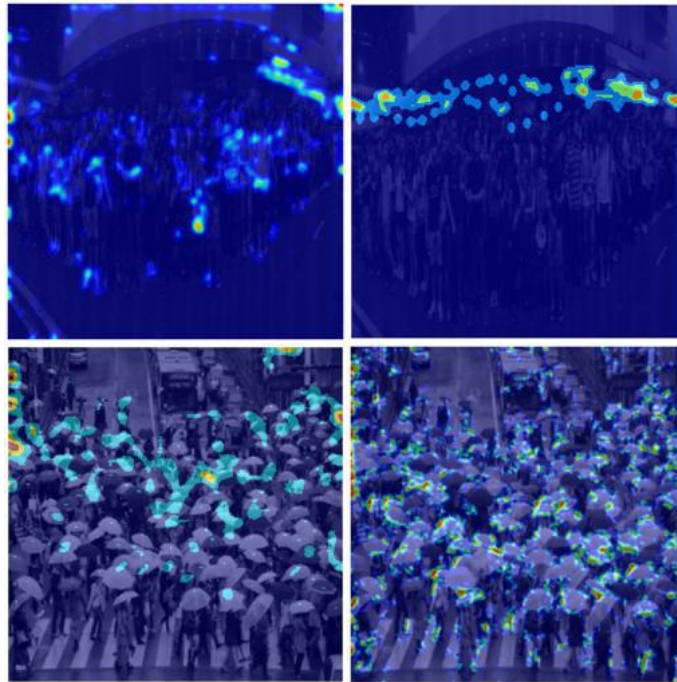


Figure 5: Density map visualization. Randomly selected two high-density samples from JHUCrowd++. The left ones are predictions, the right ones are labeled density maps.

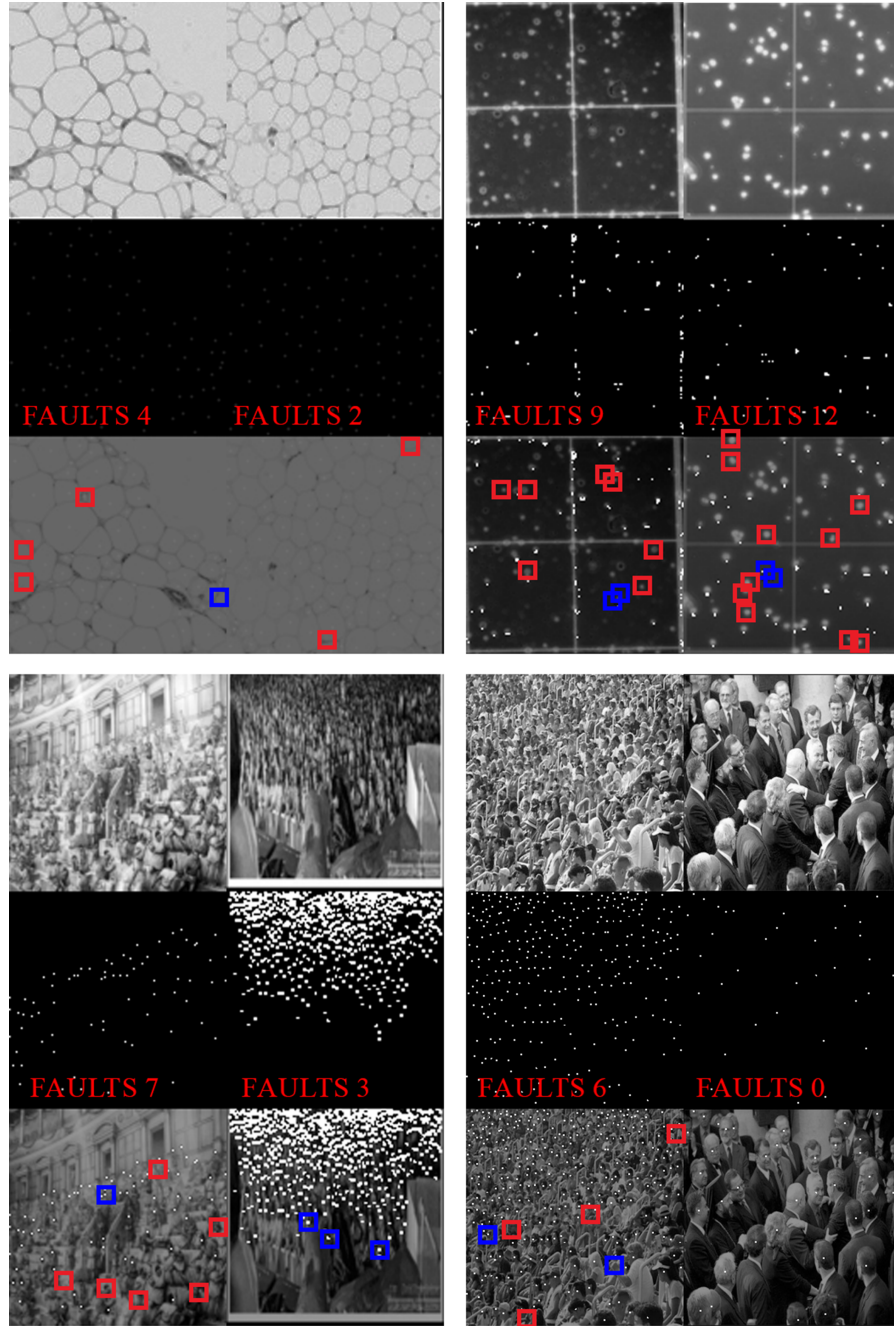


Figure 6: Dot map visualization. Randomly selected eight low-density samples from two adaptation tasks. From left to right, the samples are from ADI, DCC, UCF, SHB. The red mark indicates the miss count. The blue mark indicates the duplicated count.

A.9 FUTURE AND LIMITATION

This paper has several limitations that can be further investigated and improved. First, the lower bound of the aforementioned joint error is not the tightest (Zhao et al., 2019). It means the tightest lower bound of joint error might be lower than the loss bound mentioned above. However, this does not explicitly result in a performance drop. Second, the conditions are the label categories in BiAN, however Theorem 4 can fit more conditions. Regarding the designed model BiAN, we observed that there exists limited precision in recognizing and localizing, and counting minor objects. This aspect can be further improved in addition to the proposed CM. Nevertheless, our work emphasizes the importance of matching the conditions during the adaptation process and provides a promising direction for future research.