We thank all reviewers for the constructive reviews as well as the insightful suggestions on future improvements. We are happy they unanimously support this paper. All comments will be addressed in revision.

R1 More recent baseline methods. Thanks for pointing out additional methods. As suggested, we perform further new experiments by plugging DAFD to [ea19a, ea19b] using their implementation. Detailed results on [ea19a] are presented in Table A. And we achieved 46.7 mIoU (improved by 1.2) on GTA->Cityscapes with [ea19b]. We also add comparisons to [24] in Table B. Due to small overlap between the experiments of two papers and no public available implementation of [24], we compare the best results on three sub-tasks of Office-31 with AlexNet. On average, DAFD achieves higher performance. All additional comparisons will be added to the revision.

Table A: Experiments on CAN [ea19a].

Methods	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
CAN	94.5	99.1	99.8	95.0	78.0	77.0	90.6
CAN + Ours	95.2	99.2	100.0	96.1	78.9	78.2	91.27 (0.67 ↑)

Table B: Comparisons to [24].

Methods	A→W	D→W	W→D	Avg.
[24]	76.0	96.7	99.6	90.8
Ours	77.2	97.9	98.5	91.2 (0.4 ↑)

R1 & R4 Comparison to [1]. Domain separation networks (DSN) [1] share the motivation with us by using 'domain specific' and 'domain shared' network components to improve domain invariant feature learning. However,

- Our method works as a plug-and-play module as demonstrated with the numerous architectures in the experiments, with no additional loss functions as in [1], which consequently introduces additional hyperparameters to tune.
- Our method introduces only hundreds of parameters to model one extra domain; while in DSN, three encoder networks and one decoder networks are required to model two domains, which introduces many times more parameters.
- The aforementioned additional costs also prevent DSN from being extended to large-scale experiments like the unsupervised image segmentation which can be however easily performed by using the proposed DAFD with no additional training objectives and neglectable parameter overheads.

Despite the remarkable simplicity, DAFD is comparable to DSN according to the performance on SVHN->MNIST. Due to limited overlap of the experiments reported in DSN with ours, we show additional comparisons in Table C by reimplementing DSN (since the code link provided with the original paper is taken down now), and we will add the discussion to the final revision and more experiments in the supplementary.

Table C: Comparisons to domain separatioon networks (DSN) with DANN as underlying method. Datasets include USPS (U), SVHN (S), MNIST (M), MNIST-M (MM), Synth Digits (SD), Synth Signs (SS), and GTSRB(G). ∗ denotes numbers obtained by our reimplementation.

Methods	M→U	U→M	S→M	M→MM	SD→S	SS→G
DSN	90.6∗	92.1∗	82.7	83.2	91.2	93.1
Ours	92.3	95.4	83.2	86.2	91.7	94.0

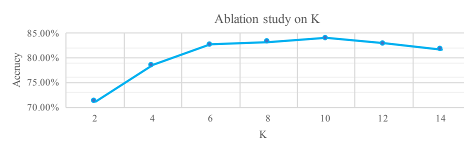

Figure 1: Ablation study on K performed on SVHN→MNIST with DANN as underlying method.

R1 & R3 Comparing to basic branching on unsupervised adaptations. Basic branching relies on heavy supervisions for every domain. In the unsupervised setting, due to the weak supervision across domains and the large number of parameters to train, the basic branching mostly cannot perform better than random guess no matter what underlying methods and initialization tricks we use. We followed the suggestion of R1 and ran experiments by equipping the domain adaptive batch normalization methods to basic branching. With several rounds of tuning, none of the methods achieved observable improvement over random guess on unsupervised experiments with basic branching. Due to the page limit, we originally removed the comparisons to basic branching in the unsupervised domain adaptation section, and we will add back the discussion in revision.

R2 Filters in real datasets. In experiments on real-world datasets, we find it challenging to interpret visual examples of learned dictionaries for domains. Thus, we bridge the gap between explainable toy examples and real-world experiments with theoretical analysis in Section 3 and supplementary material. We will visualize some of the filters and dictionaries in the supplementary.

R1 & R3 Implementation details. We use $K = 6$ as mentioned in L135. We add the ablation study on K here, which shows that DAFD is only sensitive to very small K, which degrades the expressiveness. We ease the hyperparameters tuning by decomposing all the convolution layers in every experiment as mentioned in L231, L285, and Section 3. Our goal is to illustrate in the experiments, that DAFD is a general plug-and-play module requiring little tuning of configurations and hyperparameters. We will clarify implementation details in the revision of supplementary material.

References

[ea19a] Guoliang Kang et al. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

[ea19b] Tuan-Hung Vu et al. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.