Semi-Supervised Domain Generalization with Stochastic StyleMatch

Kaiyang Zhou **Chen Change Loy** **Ziwei Liu**
S-Lab, Nanyang Technological University, Singapore
{kaiyang.zhou, ccloy, ziwei.liu}@ntu.edu.sg

Abstract

We study semi-supervised domain generalization (SSDG), a more realistic problem setting than existing domain generalization research. In particular, SSDG assumes only a few data are labeled from each source domain, along with abundant unlabeled data. Our proposed approach, called StyleMatch, extends FixMatch's two-view consistency learning paradigm in two crucial ways to address SSDG: first, stochastic modeling is applied to the classifier's weights to mitigate overfitting in the scarce labeled data; and second, style augmentation is integrated as a third view into the multi-view consistency learning framework to enhance robustness to domain shift. Two SSDG benchmarks are established where StyleMatch outperforms strong baseline methods developed in relevant areas including domain generalization and semi-supervised learning. The source code is released at https://github.com/KaiyangZhou/ssdg-benchmark.

1 Introduction

Most existing domain generalization (DG) methods assume data obtained from different source domains are fully annotated [4, 11, 12, 10, 1, 3, 14, 22, 23, 20, 21]. In this work, we turn to a more realistic and practical setting called *semi-supervised domain generalization* (SSDG), where only a few source data are labeled while the majority of them are unlabeled. The goal is to design a data-efficient DG algorithm that can utilize the unlabeled source data while coping with heterogeneity caused by different sources that further increases the difficulty of the problem.

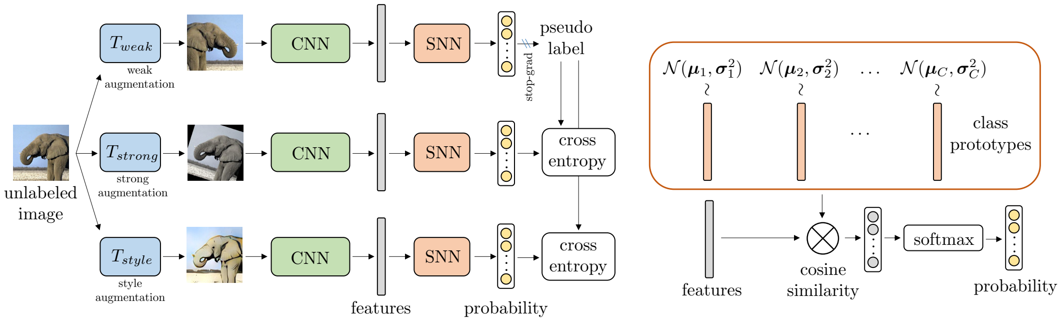
In this paper, we propose a simple yet effective approach called StyleMatch, which extends FixMatch [15] by i) introducing uncertainty to the classifier's weights—to reduce overfitting—and ii) integrating style augmentation into the multi-view consistency learning framework. For evaluation, we establish two SSDG benchmarks based on two widely used DG datasets and include a wide range of strong baseline methods developed for domain generalization and semi-supervised learning for comparison. The results demonstrate that StyleMatch achieves the best out-of-distribution generalization performance.

2 Our Approach

Our approach, StyleMatch, consists of two key components: a multi-view consistency learning paradigm and a stochastic classifier, which are sketched in Figure 1(a) and (b) respectively. Below we talk about these two components in more detail.

Multi-view Consistency Learning StyleMatch is built on top of FixMatch [15], a SOTA semi-supervised method based on pseudo-labeling: the predictions made on strongly augmented data should match the pseudo labels predicted using the weakly augmented counterparts (see the top-two streams in Figure 1(a)). To enhance domain-generalizable feature learning, we add a third

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).



(a) Multi-view consistency learning

(b) Stochastic neural network (SNN) classifier

Figure 1: Two core components in StyleMatch.

complementary view based on style transfer [7], which is shown in the bottom stream in Figure 1(a). This design is motivated by the observation that image style is closely related to visual domain [23]; and by generating structural patterns through style transfer, we can complement strong augmentation that only covers geometrical and color intensity transformations. During training, we randomly sample images from all source domains and apply AdaIN [7] to map images from one domain to another.

Stochastic Classifier To reduce overfitting on the scarce labeled source data, we build a stochastic classifier that models the classifier's weights using Gaussian distributions, shown in Figure 1(b). In doing so, instead of optimizing a fixed set of weights, we optimize their probability distributions, essentially learning an ensemble of classifiers. Formally, let $\boldsymbol{W} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_C]^T$ denote a set of weight vectors for C classes—each seen as a class prototype—and $\boldsymbol{z}$ the features for an image $\boldsymbol{x}$, the matrix-vector multiplication of $\boldsymbol{W}\boldsymbol{z}$ essentially computes the (cross-correlation) similarity between the image $\boldsymbol{x}$ and each class prototype $\boldsymbol{w}_c$ with $c \in \{1, ..., C\}$. In the stochastic classifier, each $\boldsymbol{w}$ is modeled using a Gaussian distribution parameterized by $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$. At each training step, we sample for each class the prototype from a probability distribution, $\boldsymbol{w}_c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$. To allow end-to-end optimization, we employ the widely used reparameterization trick [8, 2] to bypass the discrete sampling process,

$$\boldsymbol{w}_c = \boldsymbol{\mu}_c + \text{softplus}(\boldsymbol{\sigma}_c) \odot \boldsymbol{\epsilon} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{1}$$

Once all class prototypes are obtained, the similarity scores can be computed based on cosine similarity (denoted by $\text{sim}(\cdot, \cdot)$), which are then passed to the softmax function for generating a normalized probability distribution,

$$p(y|\boldsymbol{x}) = \frac{\exp(\text{sim}(\boldsymbol{z}, \boldsymbol{w}_y)/\tau)}{\sum_{c=1}^{C} \exp(\text{sim}(\boldsymbol{z}, \boldsymbol{w}_c)/\tau)}, \tag{2}$$

where τ is a temperature hyper-parameter, which is fixed to 0.05. At test time, we simply use the mean parameters of the probability distributions (i.e. $\boldsymbol{w}_c = \boldsymbol{\mu}_c$) to classify images in a deterministic manner.

3 Experiments

SSDG Benchmarks We repurpose two widely used DG datasets, PACS [9] and OfficeHome [17], for benchmarking SSDG methods. These two datasets focus on image classification. PACS consists of four distinct domains—art painting, cartoon, photo, and sketch—and contains 9,991 images of 7 classes in total. The domain shift is mainly concerned with image style changes. OfficeHome also has four domains: art, clipart, product, and real world. It contains more images than PACS: around 15,500 images of 65 classes, which are related to office and home objects, such as computer, chair, and bed.

Evaluation Metrics The common leave-one-domain-out protocol is adopted: three domains are used as the sources while the remaining one as the target. A model is first trained using *only the source data*, and then directly deployed in the target domain. Top-1 accuracy is used as the performance

Table 1: Domain generalization results in the low-data regime on PACS (averaged over 5 random splits). A: Art painting. C: Cartoon. P: Photo. S: Sketch. $\boldsymbol{u}$: use unlabeled source data.

Model	$\boldsymbol{u}$	# labels: 210 (*10 per class*)					# labels: 105 (*5 per class*)				
		A	C	P	S	*Avg*	A	C	P	S	*Avg*
Full-Labels	-	76.95	75.90	95.96	69.20	*79.50*	76.95	75.90	95.96	69.20	*79.50*
Domain generalization methods											
Vanilla	✗	63.09	58.49	86.56	45.56	*63.42*	56.71	53.87	71.87	36.92	*54.84*
CrossGrad [14]	✗	62.56	58.92	85.81	44.11	*62.85*	56.39	55.11	72.61	38.08	*55.55*
DDAIG [22]	✗	61.95	58.74	84.44	47.48	*63.15*	55.09	52.31	70.53	38.89	*54.20*
MixStyle [23]	✗	71.11	64.04	88.99	54.62	*69.69*	62.00	58.40	80.43	43.58	*61.10*
EISNet [18]	✓	66.84	61.33	89.16	51.38	*67.18*	62.08	54.75	80.66	42.68	*60.04*
Semi-supervised learning methods											
MeanTeacher [16]	✓	62.41	57.94	85.95	47.66	*63.49*	56.00	52.64	73.54	36.97	*54.79*
EntMin [5]	✓	72.77	70.55	89.39	54.38	*71.77*	67.01	65.67	79.99	47.96	*65.16*
FixMatch [15]	✓	78.01	68.93	87.79	73.75	*77.12*	77.30	68.67	80.49	73.32	*74.94*
Semi-supervised domain generalization methods											
StyleMatch (*ours*)	✓	**79.43**	**73.75**	**90.04**	**78.40**	***80.41***	**78.54**	**74.44**	**89.25**	**79.06**	***80.32***

Table 2: Domain generalization results in the low-data regime on OfficeHome (averaged over 5 random splits). A: Art. C: Clipart. P: Product. R: Real world. $\boldsymbol{u}$: use unlabeled source data.

Model	$\boldsymbol{u}$	# labels: 1950 (*10 per class*)					# labels: 975 (*5 per class*)				
		A	C	P	R	*Avg*	A	C	P	R	*Avg*
Full-Labels	-	58.88	49.42	74.30	76.21	*64.70*	58.88	49.42	74.30	76.21	*64.70*
Domain generalization methods											
Vanilla	✗	50.11	43.50	65.11	69.65	*57.09*	45.76	39.97	60.04	63.77	*52.38*
CrossGrad [14]	✗	50.32	43.27	65.16	69.49	*57.06*	45.68	40.04	59.95	64.09	*52.44*
DDAIG [22]	✗	49.60	42.52	63.54	67.89	*55.89*	45.73	38.82	59.52	63.37	*51.86*
MixStyle [23]	✗	49.79	47.12	64.18	68.42	*57.38*	46.51	43.59	59.66	63.30	*53.26*
EISNet [18]	✓	51.16	43.33	64.72	68.36	*56.89*	47.32	40.07	59.33	62.59	*52.33*
Semi-supervised learning methods											
MeanTeacher [16]	✓	49.92	43.42	64.61	68.79	*56.69*	44.65	39.15	59.18	62.98	*51.49*
EntMin [5]	✓	51.92	44.92	**66.85**	**70.52**	*58.55*	48.11	41.72	**62.41**	63.97	*54.05*
FixMatch [15]	✓	50.36	49.70	63.93	67.56	*57.89*	48.98	47.46	60.70	64.36	*55.38*
Semi-supervised domain generalization methods											
StyleMatch (*ours*)	✓	**52.82**	**51.60**	65.31	68.61	***59.59***	**51.53**	**50.00**	60.88	**64.47**	***56.72***

measure. Two SSDG settings are designed. In the first setting, we randomly sample 10 images per class from each source domain and use the rest as unlabeled data. The second setting tests a much more challenging scenario: only five labeled images are available for each class in each source domain. Results are averaged over five random splits.

Training Details The ImageNet-pretrained ResNet18 [6] is used as the CNN backbone. We randomly sample 16 images from each source domain to construct a minibatch, for labeled and unlabeled data, respectively. Following FixMatch, the labeled minibatch is used for computing the labeled loss, while both labeled and unlabeled minibatches are used to compute the two unlabeled losses. The initial learning rate is set to 0.003 for the pretrained backbone and 0.01 for the randomly initialized stochastic classifier, both decayed by the cosine annealing rule. The number of training epochs is 40 for PACS and 20 for OfficeHome. We use a single Tesla V100 GPU for model training. Our implementation is based on the public `Dassl.pytorch` toolbox.[1]

[1] `https://github.com/KaiyangZhou/Dassl.pytorch`.

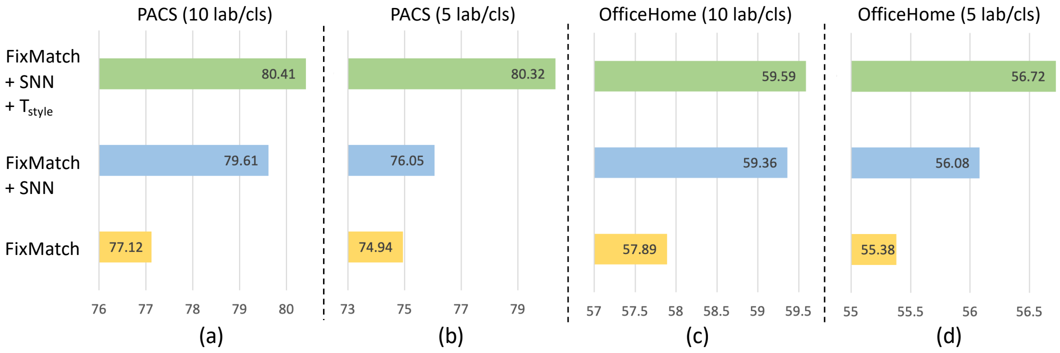


Figure 2: Ablation study on two key components in StyleMatch: the stochastic neural network (SNN) classifier and the style augmentation T_{style}.

Results on PACS The results are presented in Table 1 where we include SOTA domain generalization (DG) and semi-supervised learning (SSL) methods for comparison. Full-Labels follows Vanilla's training strategy but uses all labels in each source domain. It can be seen that most DG methods do not work well given the limited labeled data. MixStyle performs exceptionally well compared to its fellows, suggesting that feature-level augmentation has a good potential in tackling label shortage. Nonetheless, the gap between MixStyle and Full-Labels is still huge. Among all DG methods, only EISNet can use the unlabeled source data due to its self-supervised loss based on Jigsaw puzzles [13], and has apparently benefited from this design. The SSL methods generally outperform the DG methods. Among them, FixMatch achieves outstanding performance. Finally, our StyleMatch demonstrates clear advantages over all baselines. It is also worth noting that only our approach suffers the smallest performance drops in the 5-shots setting.

Results on OfficeHome The results are shown in Table 2. The gaps between different methods are generally smaller than those on PACS—this is because OfficeHome has more classes and thus poses more challenges. Therefore, a small increase in the accuracy on OfficeHome would be appreciated. The observations are similar to those on PACS: most DG methods fail to beat Vanilla and the SSL methods are generally better. StyleMatch again achieves the best overall out-of-distribution generalization performance. Nonetheless, the gaps with Full-Labels are noticeable, suggesting that there is still room for improvements for future work.

Ablation Study We conduct a comprehensive ablation study to examine the effectiveness of the two proposed components: the SNN classifier and the style augmentation T_{style}. We repeat the experiments on PACS and OfficeHome by sequentially adding these two components to FixMatch. Figure 2 shows the results of this ablation study with a focus on the average accuracy over all target domains. We observe that SNN contributes around 2% and 1% increase to the performance on PACS and OfficeHome, respectively, and T_{style} further boosts the performance. In particular, by adding T_{style} to FixMatch+SNN, the improvements obtained are higher in the lower-data setting on both datasets, suggesting that T_{style} (multi-view consistency learning) is essential when dealing with extremely scarce labels.

We leave more analytical studies in Appendix A.1, including the effect of SNN, complementarity in augmentation methods and the impact of number of sources.

4 Conclusion

Semi-supervised domain generalization, a more realistic and practical setting, greatly challenges the design of existing DG methods. We show that with limited labels the previous top-performing DG methods fail to learn generalizable representations, while our specifically designed StyleMatch gains huge improvements in reducing the gap with full-labels training. Nonetheless, we observe that pseudo labels' accuracy is much lower in our setting compared to that in a traditional semi-supervised setting where data are sampled from a single distribution. Therefore, future work can be focused on designing new formulations to incorporate source domain shift for building more robust semi-supervised domain generalization algorithms.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015.

[3] Qi Dou, Daniel C Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019.

[4] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *TPAMI*, 2017.

[5] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2004.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[9] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[10] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

[11] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018.

[12] Ya Li, Xinmei Tiana, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018.

[13] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[14] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

[15] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[16] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[17] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[18] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *ECCV*, 2020.

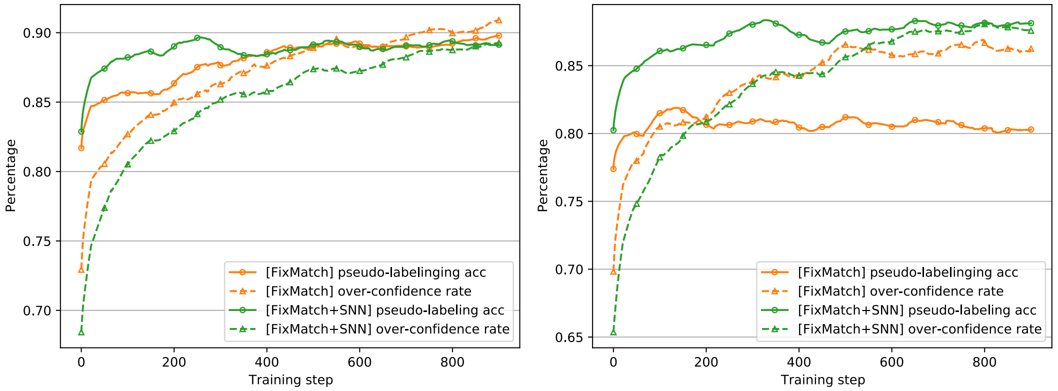
[19] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[20] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[21] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020.

[22] Kaiyang Zhou, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020.

[23] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.



(a) The **10-labels-per-class** setting on PACS with art painting as the target

(b) The **5-labels-per-class** setting on PACS with art painting as the target

Figure 3: Pseudo-labeling accuracy (*solid+circle*) vs. over-confidence rate (*dashed+triangle*). Without the SNN-based classifier, the model suffers from severe overfitting, which is reflected in the over-confidence rate overshooting the pseudo-labeling accuracy.

Table 3: Strong vs. style augmentation.

	PACS	
StyleMatch's variants	10 lab/cls	5 lab/cls
T_{strong}-only	79.61	76.05
T_{style}-only	72.61	69.72
$T_{strong}+T_{style}$	**80.41**	**80.32**

Table 4: Impact on number of sources (K).

	PACS					
	10 lab/cls			5 lab/cls		
	$K=1$	$K=2$	$K=3$	$K=1$	$K=2$	$K=3$
FixMatch	53.55	71.42	77.12	49.91	68.52	74.94
StyleMatch	**57.29**	**74.50**	**80.41**	**52.24**	**71.95**	**80.32**

A Appendix

A.1 Further Analysis

Stochastic Classifier Reduces Overfitting To understand how the stochastic classifier improves learning, we compare FixMatch+SNN with FixMatch using two metrics: pseudo-labeling accuracy and over-confidence rate, which are measured for each minibatch data received at each training step. The first metric measures the accuracy of pseudo labels, while the over-confidence rate counts how many pseudo labels in a minibatch pass the confidence threshold. Ideally, we do not want the over-confidence rate to climb above the pseudo-labeling accuracy as this would mean the network predicts excessive incorrect pseudo labels with high confidence, which hurt generalization [19]. Figure 3 shows the comparisons. In (a), the over-confidence rate of FixMatch+SNN steadily increases and eventually converges to a similar level as the pseudo-labeling accuracy. In contrast, without SNN, the over-confidence rate overshoots the pseudo-labeling accuracy in the middle of training. In (b), the overfitting issue for FixMatch intensifies—the over-confidence rate outpaces the pseudo-labeling accuracy at the early training stage and the pseudo-labeling accuracy stops improving since then. In contrast, the curves for FixMatch+SNN look much healthier.

Complementarity in Augmentation Methods To provide an in-depth analysis of the role of augmentation methods, we evaluate three variants of StyleMatch: T_{strong}-only, T_{style}-only, and $T_{strong}+T_{style}$. T_{strong}-only and T_{style}-only are based on the two-view consistency learning paradigm and $T_{strong}+T_{style}$ refers to the final model. Table 3 shows the results of this ablation study on PACS. We observe that 1) T_{strong} is more suitable than T_{style} to be used in the two-view consistency learning framework, and 2) combining these two augmentation methods leads to a much better performance, which justifies their complementarity.

Impact on Number of Sources The previous experiments use three sources. To investigate the impact on the number of sources, we further conduct experiments by reducing the number of sources from three to two/one. For each target domain, the experiments cover all possible scenarios with different combinations of sources, each following the five random splits. For example, when sketch is used as the target and the number of sources is set to two, there are three different scenarios: 1) art painting and cartoon as the sources, 2) art painting and photo as the sources, and 3) cartoon and photo as the sources. The average accuracy is shown in Table 4. Note that when $K = 1$, we mix image style between random instances from the same domain for StyleMatch. The results demonstrate that StyleMatch outperforms FixMatch in all scenarios, even in the single-source case—this means mixing instance-level style also helps, which is consistent with the observation in a recent work that mixes instance-level feature statistics [23]. By increasing K from 2 to 3, StyleMatch gains 5.91% (from 74.50% to 80.41%) and 8.37% (from 71.95% to 80.32%) respectively in the 10- and 5-labels-per-class settings, while FixMatch's gains are 5.7% (from 71.42% to 77.12%) and 6.42% (from 68.52% to 74.94%) respectively, which are smaller than those of StyleMatch. Therefore, StyleMatch benefits more from having more sources, which makes sense: T_{style} can generate more diverse images given more sources.