AGKD-BML: Defense Against Adversarial Attack by Attention Guided Knowledge Distillation and Bi-directional Metric Learning — Supplementary Materials —

1. Evaluation on Tiny ImageNet

We evaluate our method on a larger dataset, *i.e.*, *Tiny ImageNet*, which is a tiny version of ImageNet consisting of 3-channel color images with size of 64×64 belonging to 200 classes. Each class has 500 training images and 50 validation images. We use the comparison methods include *Undefended Model (UM)*, *adversarial training (AT)* [5], *adversarial logit pairing (ALP)* [4], *single directional metric learning (SML)* [6], *Bilateral* [8] and *feature scattering (FS)* [11]. To reduce the computational cost, we use ResNet-50 model as the same as SML. The learning rate γ is initialized as 0.1, and decays at 30 epoch. We retrain and evaluate the models of bilateral and feature scatter using the existing codes.

From Table 1, we can see that all methods show relatively poor performance on Tiny ImageNet. While our method outperforms others by a small margin, our performance can only achieve $\sim$20%, which is not good enough for practical usage. It suggests that Tiny ImageNet is a difficult dataset due to its large class numbers and small sample size in each class. There is a large room to improve on this difficult dataset.

2. Additional Results of AGKD-BML Model Against AutoAttack (AA) [3]

In Table 2, we provide additional results of AGKD-BML model trained on 10-step attacks against AutoAttack (AA) [3] which is an ensemble of four diverse attacks. We compare two Wide ResNet [10] structures, *i.e.*, WRN-28-10 and WRN-34-10, as well as two different learning rate decay epochs, *i.e.*, 100 and 150. For our AGKD-BML models trained with large-number-step attacks, we utilize the MART loss [9] which explicitly emphasizes misclassified examples. Following the suggestions in [7, 9] that the best performance is usually on a few epochs after the first learning rate decay, we stop our training at 5 epochs after the first learning rate decay. From Table 2, we can see that with more layers, *i.e.* 34 v.s. 28, the model usually performs better in terms of the accuracy against AA.

3. Class-irrelevant attention distillation

In our work, we use the class-*irrelevant* attention to transfer the regions that the model focuses on, regardless of which class makes the contribution. By contrast, the class-*relevant* attention map shows the attention region related to a specific class. An adversarial example (AE) fools a neural network (NN) by adding intentionally designed perturbations, which are further augmented by the NN to make the values of false class (actual prediction on AE) related attention map surpass that of original class, and thus make the NN misclassify the sample. We argue that transferring the information of class-relevant attention map is problematic:
1) *For the original class, transferring the class relevant attention* has limited effects since AE hurts much less the original class attention map than the false class one. More importantly, it rarely reduces the dominated responses of the false class related maps which limits the effects of correcting the misclassification.
2) *For the false class, transferring the class relevant attention* enforces the false class attention map to focus on the regions where objects of the original class locate, and it does not modify the attention regions of the original class.

We conducted the experiment to compare the class relevant/irrelevant attention distillation. We trained models by the class-relevant attention maps generated by Grad-CAM corresponding to both original and false classes. In addition to our AGKD, we also trained a model by another class irrelevant attention map generated by averaging all CAM maps of all classes [12]. The results in Table 3 demonstrate better performance achieved by the class-irrelevant attention distillation. We chose AGKD over CAM-avg since CAM-avg is computationally heavy.

4. Comparison of Running Time

We provide the training time of bilateral [8], feature scatter [11], AT [5], SML [6] and our AGKD-BML model on CIFAR-10 dataset. In Table 4, we provide implementation platforms, training time (seconds) per epoch, number of epochs for training, total time (hours) for training, as well as the number of the steps to get the adversarial examples for

Table 1. Evaluation results on Tiny ImageNet, under seven widely used attacks, as well as the results on clean images. The best accuracy for each attack is illustrated as bold. All attack budgets in training are $\epsilon = 8$ by default for an apples to apples comparison.

Tiny ImageNet								
Attacks(steps)	clean	FGSM	BIM(10)	PGD (10)	PGD (20)	CW (10)	CW (20)	MIM (40)
UM	**64.62**%	3.93%	0.17%	0.10%	0.07%	0%	0%	0.57%
Bilateral [8]	58.70%	30.81%	20.98%	19.73%	18.98%	15.19%	14.61%	22.47%
FS [11]	53.81%	30.06%	20.59%	19.46%	18.52%	15.53%	14.68%	22.40%
AT [5]	42.29%	26.08%	20.41%	19.99%	19.59%	17.17%	16.92%	21.15%
ALP [4]	41.53%	21.53%	20.03%	20.18%	19.96%	16.80%	-	19.85%
SML [6]	40.89%	22.12%	20.77%	20.89%	20.71%	17.48%	-	20.69%
AGKD-BML	53.21%	**31.39**%	**23.55**%	**22.68**%	**21.78**%	**18.8**%	**18.03**%	**24.71**%

Table 2. Evaluation results of AGKD-BML against AutoAttack (AA), on CIFAR-10 dataset, with different layers, learning rate and decay points. All results are evaluated by the models trained on 10-step attacks.

CIFAR-10			
	Networks	Decay point	AA
AGKD-BML-10	WRN-28-10	100	50.80%
		150	50.73%
	WRN-34-10	100	51.05%
		150	51.63%

Table 3. Comparisons of class-relevant and -irrelevant attention.

Class-relevant		Class-irrelevant	
GC-orig	GC-false	CAM-avg	AGKD
59.06%	58.86%	66.55%	65.93%

each model. All running times are evaluated on one Nvidia V100 GPU with 32GB memory. Once trained, testing times for all the models are approximately the same, although it shows in Table 4 that our model takes more time in training. In the security applications, the training time is not critical compared to the accuracy. Therefore, 1.2 days of training time of AGKD-BML model is acceptable.

Furthermore, we also evaluated the performance of different models with the same running time. FS and Bilateral originally use only one-step attacks. Therefore, we trained FS with 2-step attack (470s) and Bilateral with 4-step attack (453s), and got accuracy of 70.51% and 59.99% (compared to ours: **71.02**%, 528s), respectively.

5. Comparison of One More Latest Model

In [2], the authors proposed a customized adversarial training (CAT) model, which adaptively tunes a suitable ϵ for each sample during the adversarial training procedure. However, the authors do not provide systematical results of the same experiment settings as that in [6, 8, 11], instead, they only provide the results under PGD and CW attacks on CIFAR-10 dataset. Therefore, we report our results of the same experimental setting as CAT in Table 5. CAT has two variants, 1) "CAT-CE" applies standard cross entropy loss as used in traditional adversarial training models [5], and 2) "CAT-MIX" applies both cross entropy loss and CW loss [1]. Note that CAT used an adaptive ϵ for training, while our results are given with a fixed $\epsilon = 4$. From the table, we can see that our proposed AGKD-BML model consistently outperformed "CAT-CE", and "CAT-MIX" except under "CW" attack. This is because both AGKD-BML and CAT-CE only apply cross entropy which is used in PGD attack, and "CAT-MIX" includes both cross entropy loss and CW loss that is used in CW attack.

6. Evaluation of k-Nearest Neighbor (k-NN) classifier

We conduct the experiments that apply k-Nearest Neighbor (k-NN) method as the classifier following [6]. We utilize the feature vectors from the penultimate layer to perform the k-NN classifier for all the models with $k = 50$.

In Table 6, we show the k-NN classifier accuracy, as well as corresponding softmax accuracy, between four models, *i.e.*, AT [5] ALP [4] SML [6] and proposed AGKD-BML, on CIFAR-10, SVHN and Tiny ImageNet datasets. Our model consistently achieves higher accuracy on all three datasets. Moreover, the k-NN classifier usually performs very similarly as softmax. These quantitative results, coupled with the visualization illustrations in next section (Section 7), demonstrate that AGKD-BML is able to obtain better representation in the latent feature space than other comparison methods, and the accuracy does benefit from the good representation, rather than the classifier.

7. Visualization Analysis with t-SNE

In Figure 1, 2, 3, 4, we provide the t-SNE plots to show the sample representations in feature space for all attacked classes under PGD-20 and PGD-100 attacks. The triangle points with different colors represent the clean images in different classes, while the red circle points are the adversarial examples under attack. We show the adversarial examples of airplane, automobile, bird, cat and deer under PGD-20 and PGD-100 attack in Figure 1 and Figure 3, respectively, and the adversarial examples of dog, frog, horse,

Table 4. Training time comparison.

CIFAR-10					
	♯ of steps	platform	seconds / epoch	♯ of epochs	total (hours)
Bilateral [8]	1	TensorFlow	211s	200	11.7h
FS [11]	1	PyTorch	342s	200	19.0h
AT [5]	7	PyTorch	502s	200	27.9h
SML [6]	7	TensorFlow	2234s	55	34.1h
AGKD-BML	2	PyTorch	528s	200	29.3h

Table 5. Comparing with customized adversarial training (CAT). Note that "CAT-MIX" applies CW as part of its loss.

CIFAR-10					
Models	White-box			Black-box	
	clean	PGD	CW	VGG-16	Wide ResNet
CAT-CE [2] (adaptive ϵ)	93.48%	73.38%	61.88%	86.58%	88.66%
CAT-MIX [2] (adaptive ϵ)	89.61%	73.16%	**71.67**%	-	-
AGKD-BML (fixed $\epsilon = 4$)	**95.04**%	**77.45**%	69.06%	**89.12**%	**91.98**%

ship and truck under PGD-20 and PGD-100 attack in Figure 2 and Figure 4, respectively.

From these figures, the same observations can be seen as in the main text, which we would like to emphasize as following:

- In the first column of all the figures, "UM" is almost *non-robust* to the adversarial examples, as it shows that all the adversarial examples are far away from their original class, and fit into the distributions of other classes.
- In the second and third columns of all the figures, while SML does pull many of the adversarial examples back to their original class, BML keeps better separations between different classes, and has much less amount of adversarial examples located far away compared to SML.
- By integrating both AGKD and BML, AGKD-BML pulls most of the adversarial examples back to their original class, while keeps best separation between classes overall, as shown in the fourth column of all the figures.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (SP)*, pages 39–57, 2017. 2

[2] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv:2002.06789*, 2020. 2, 3

[3] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pages 2206–2216, 2020. 1

[4] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv:1803.06373*, 2018. 1, 2, 4

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 3, 4

[6] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *NeurIPS*, pages 480–491, 2019. 1, 2, 3, 4

[7] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, pages 8093–8104. PMLR, 2020. 1

[8] Jianyu Wang and Haichao Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *ICCV*, pages 6629–6638, 2019. 1, 2, 3

[9] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2019. 1

[10] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, pages 87.1–87.12, 2016. 1

[11] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *NeurIPS*, pages 1831–1841, 2019. 1, 2, 3

[12] Bolei Zhou et al. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 1

Table 6. Evaluation and comparison between k-NN and softmax classifiers (k-NN/softmax). Our proposed AGKD-BML with k-NN classifier consistently outperforms other comparison methods, and shows very similar accuracy with softmax classifier.

Models	CIFAR-10		SVHN		Tiny ImageNet	
	clean	PGD (20)	clean	PGD (20)	clean	PGD (20)
AT [5]	87.1% / 86.2%	47.5% / 45.6%	91.5% / 91.6%	45.8% / 45.6%	36.6% / 42.3%	20.2% / 19.6%
ALP [4]	89.6% / 89.8%	48.9% / 48.5%	91.4% / 91.3%	52.0% / 52.2%	35.2% / 41.5%	20.3% / 20.0%
SML [6]	86.3% / 86.2%	51.7% / 51.6%	84.3% / 84.0%	52.0% / 51.9%	34.0% / 40.6%	20.7% / 20.7%
AGKD-BML	**91.9**% / **92.0**%	**71.1**% / **71.0**%	**95.1**% / **95.0**%	**75.1**% / **74.9**%	**51.8**% / **53.2**%	**21.0**% / **22.7**%

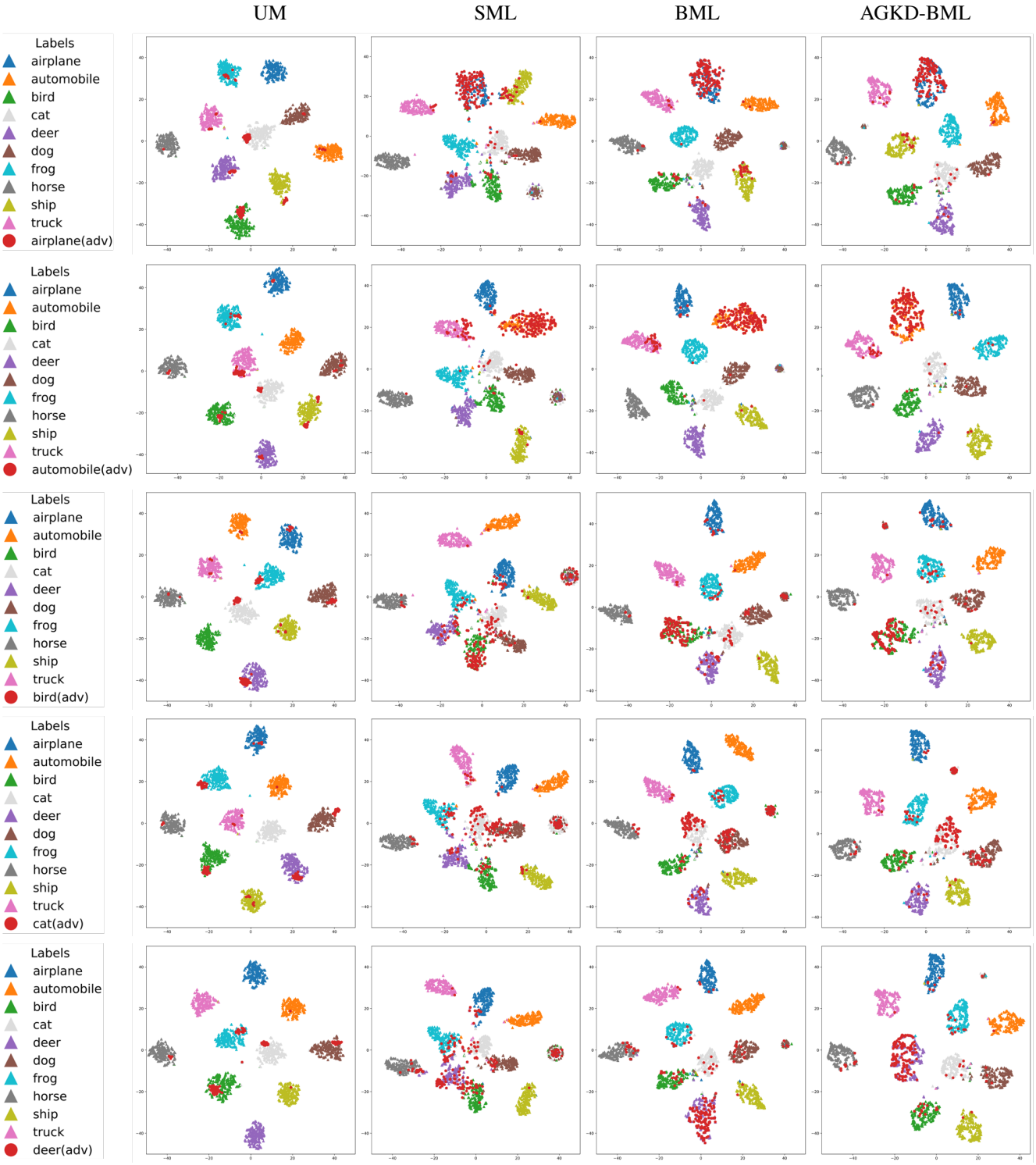

Figure 1. t-SNE plots for illustrating the sample representations in feature space. The adversarial example is airplane, automobile, bird, cat and deer in each row, respectively, under PGD-20 attack.

.

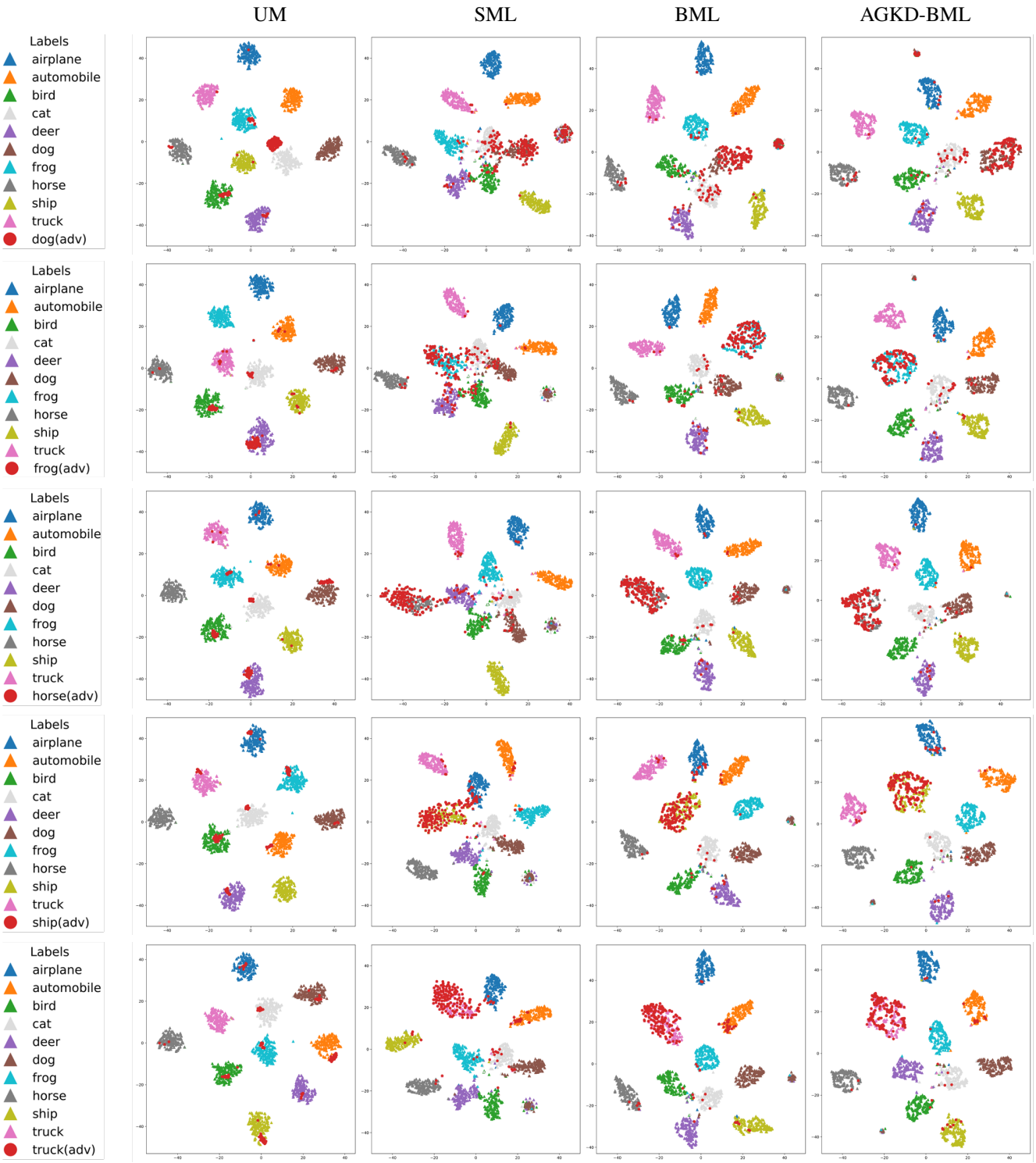

Figure 2. t-SNE plots for illustrating the sample representations in feature space. The adversarial example is dog, frog, horse, ship and truck in each row, respectively, under PGD-20 attack.

.

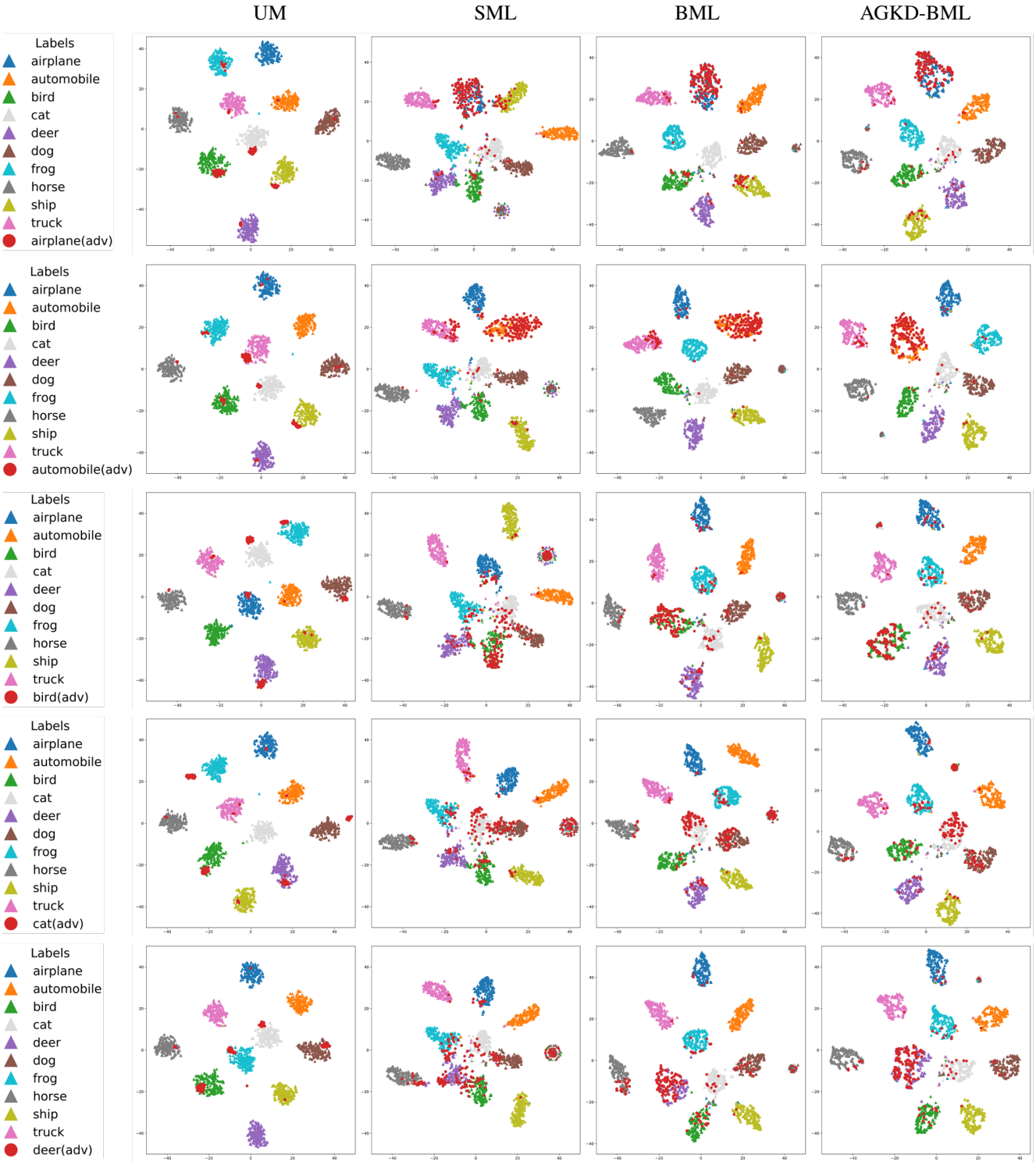

Figure 3. t-SNE plots for illustrating the sample representations in feature space. The adversarial example is airplane, automobile, bird, cat and deer in each row, respectively, under PGD-100 attack.

.

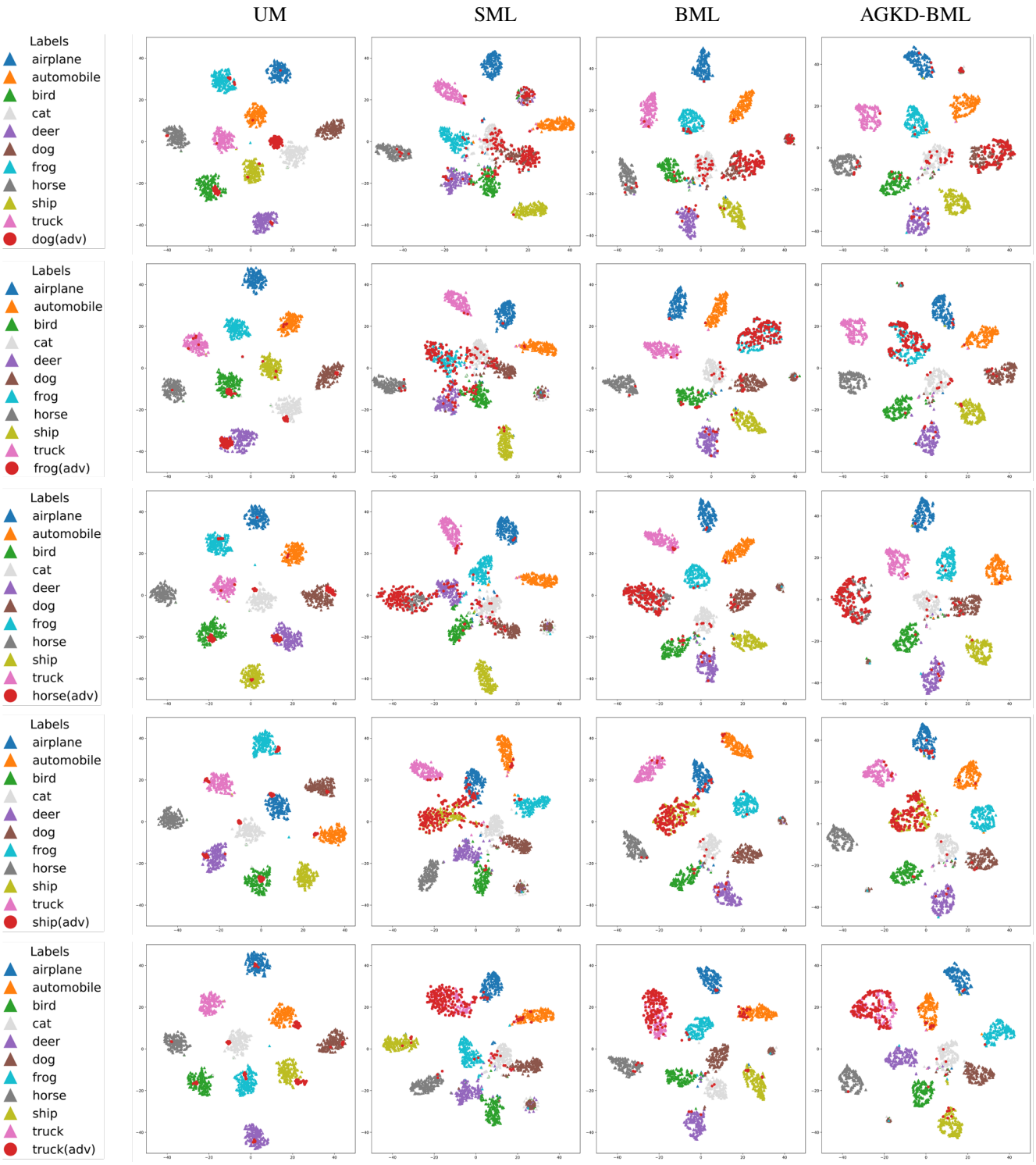

Figure 4. t-SNE plots for illustrating the sample representations in feature space. The adversarial example is dog, frog, horse, ship and truck in each row, respectively, under PGD-100 attack.

.