— Supplementary Material —

Appendix

In this appendix, we provide supplementary technical details and experiments that could not fit within the main manuscript. We present detailed information about all reference datasets used, encompassing training and validation samples, in Section 1. All the information about the CNN model training procedure, as well as details about all the hyperparameters used during training and validation, is shown in Section 2. Lastly, we present qualitative visualization results, encompassing synthetic images generated by the AGA , GradCam visualization heatmaps for enhanced explainability, and UMAP plots depicting feature clusters to assess the quality of generated image features in Section 3 .

1. Additional Dataset Details

In this section, we present additional details about all the representative datasets we used to evaluate our proposed method AGA . We use the ImageNet10 dataset, which is a subset of the original ImageNet dataset [1] with 10 different classes. These are chickadee (*n01592084*), water ouzel (*n01601694*), loggerhead (*n01664065*), box turtle (*n01669191*), garter snake (*n01735189*), sea snake (*n01751748*), black and gold garden spider (*n01773157*), tick (*n01776313*), ptarmigan (*n01796340*), prairie chicken (*n01798484*). We use the training and validation sets from ImageNet [1] for these 10 classes. We also utilize the iWildCam [2] dataset, which contains a large collection of global camera trap images of 7 different classes of background, elephant, impala, cattle, zebra, dik-dik, and giraffe, and the CUB [3] dataset, a fine-grained classification set of 200 bird species from Flickr. We maintain the same data distribution ratio as in the previous work [4] for the train and test sets to ensure a fair comparison. To show the robustness and generalization capability of our method, we additionally use two other datasets named ImageNet-Sketch [5] and ImageNet-V2 [6], where ImageNet-Sketch is the sketch version and ImageNet-V2 is the reproduced version of ImageNet. We utilize ImageNet-Sketch and ImageNet-V2 to validate the robustness of AGA against out-of-distribution samples. The number of training and validation images used for evaluation is presented in Table 1.

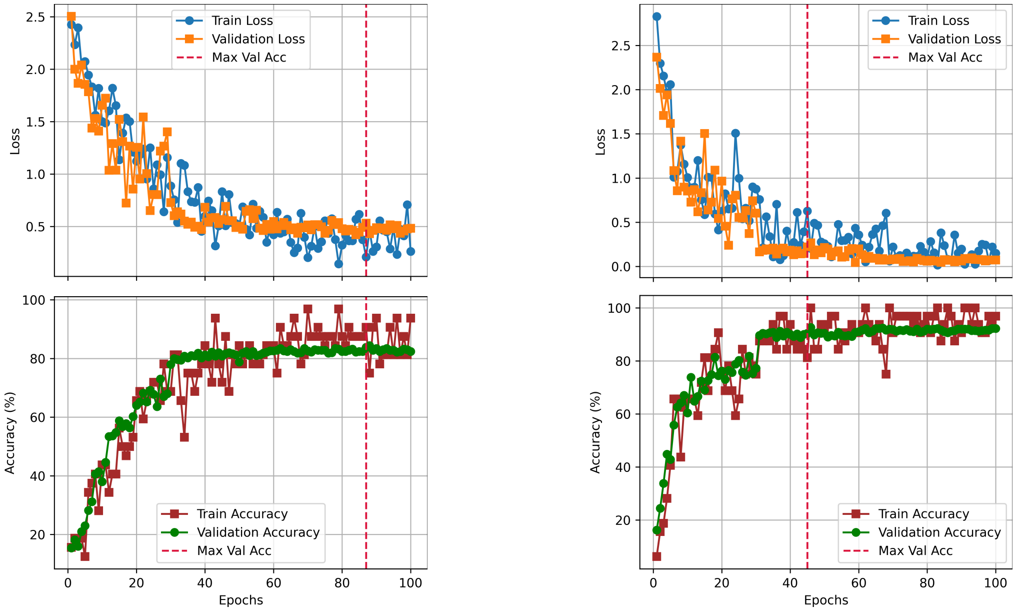
Table 1. Number of train and validation samples for all the representative datasets utilized in *AGA* . We use '-' to denote the absence of training samples on the ImageNet-Sketch and ImageNet-V2 datasets. These datasets are used for out-of-distribution validation to assess the robustness of the AGA framework.

Dataset Name	**No. of Images**	
	Training	**Validation**
ImageNet10 [1]	13046	500
iWildCam [2]	6052	8483
CUB [3]	4994	5794
ImageNet-Sketch [5]	-	511
ImageNet-V2 [6]	-	102

2. Training and Hyperparameter Details

Our automatic image augmentation framework AGA starts by separating the main subjects in the image using segmentation methods. Then, it uses a large language model (LLM) to generate different captions of backgrounds. These captions are fed into a vision model like Stable Diffusion to create various backgrounds. In the end, AGA combines the separated subjects with the newly created backgrounds. We utilize a Llama-2-13B-GPTQ from Hugging-Face [7] to create background image captions and Stable Diffusion XL [8] text-to-image model to generate background image, with default hyperparameters.

After the generation of augmented images we evaluate the quality of the additional data samples using several CNN classifier models. We employ ResNet variants 18, 50, 101, 152 as the classification models for training. We train these CNN models from scratch using PyTorch's standard training script [9] which includes PyTorch's default hyperparameter set [10]. All the hyperparameter values used for CNN classifier training are presented in Table 2. We train all the classifier models multiple times and report the average performance. While training these CNN models, we carefully addressed the issue of overfitting. We often refer to the maximum classifier accuracy for any epoch by avoiding overfitting. We ensure this by using the training and validation loss. We train all the models in such a way that the difference between the training and validation loss is minimized. The training and validation losses of ImageNet10



(a) Base model training

(b) AGA model training

Figure 1. Training loss, validation loss, and accuracy curve for the base model for real data samples and the AGA model for augmented datasamples of ImageNet 10. The training and validation losses exhibit downward trends for both the base and AGA model training, with the validation loss remaining relatively stable and not showing a significant upward trend. This indicates that the models are generalizing well to unseen data and do not show signs of overfitting.

training are presented in Figure 1, with real data shown in 1a and augmented data in 1b. The x-axis represents the number of epochs, and the y-axis represents both accuracy and loss values. In both cases, the red vertical dashed line represents the epoch at which we achieve the maximum validation accuracy. We observe that both training and validation losses exhibit downward trends in Figure 1 for both base and AGA model training. No such scenario is detected where training loss keeps decreasing while validation loss starts to increase. The validation loss remains relatively stable and doesn't show a significant upward trend. This indicates the model is generalizing well to unseen data and clearly shows no signs of overfitting.

3. Additional Results

We present additional synthetic images generated by AGA in Section 3.1. We also exhibit more GradCam visualization results to demonstrate the improved explainability of the classifier model trained with augmented data samples compared to one trained with only real samples in Section 3.2. Moreover, we display the CNN model-extracted features in a UMAP plot, showing feature clusters for different classes of real and augmented images in Section 3.3.

3.1. Additional Synthetic Images

We present more generated images from real image with diverse backgrounds. Figure 2, 3 and 4 display multiples synthetic images generated by AGA using ImageNet10 and CUB traing image samples.

Table 2. Training Details of ResNet Models

Model Parameter	**ResNet-{18,50,101,152}**
Epochs	100
Batch Size	32
Optimizer	Stochastic Gradient Descent (SGD)
Momentum	0.9
Learning Rate	0.1
Learning Rate Scheduler	StepLR
Learning Step Size	30
Gamma Parameter	0.1
Weight Decay	1e-4
Interpolation	Bilinear
Loss Function	CrossEntropyLoss

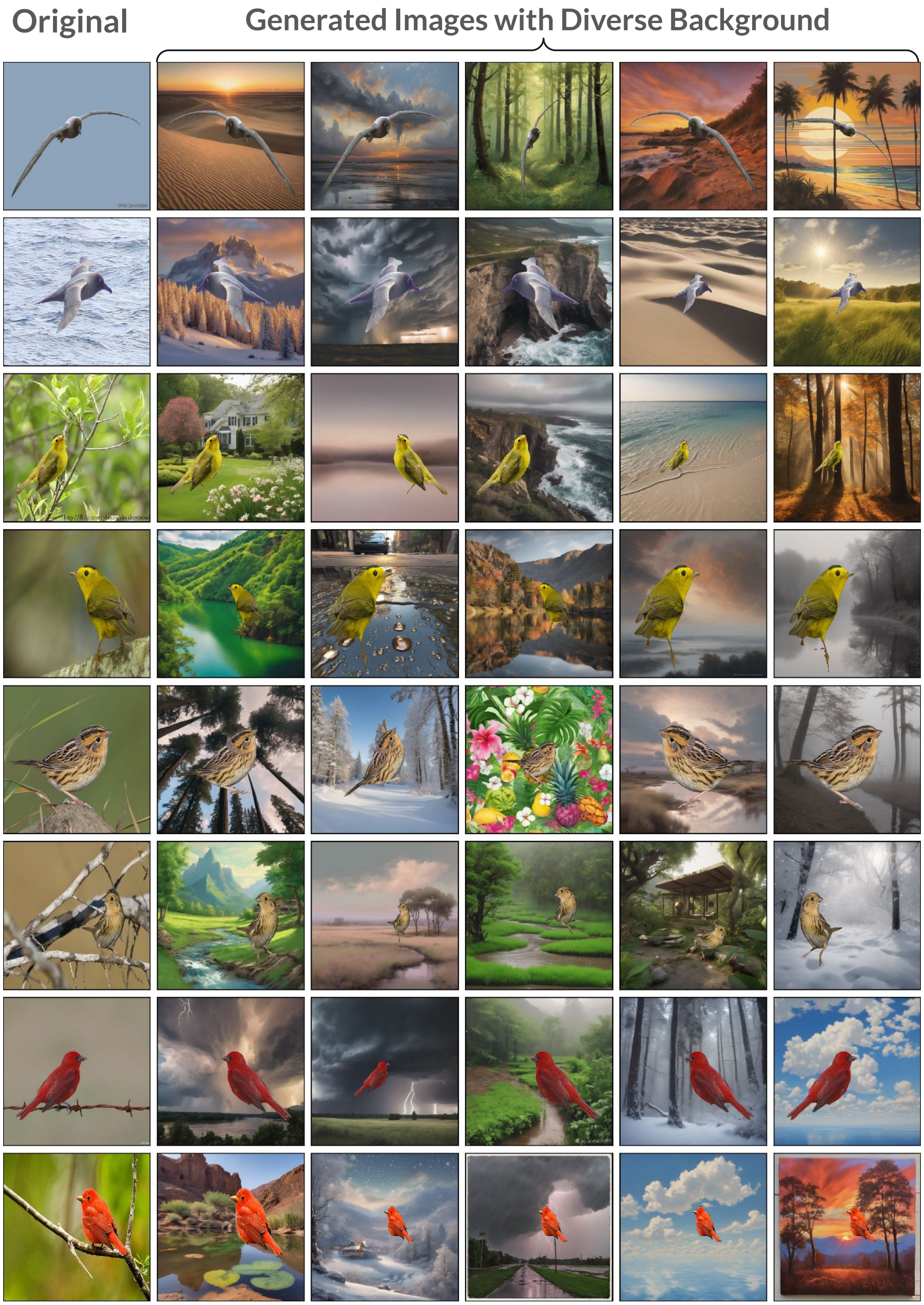


Figure 2. The figure displays the original image samples from CUB and the generated images using AGA .

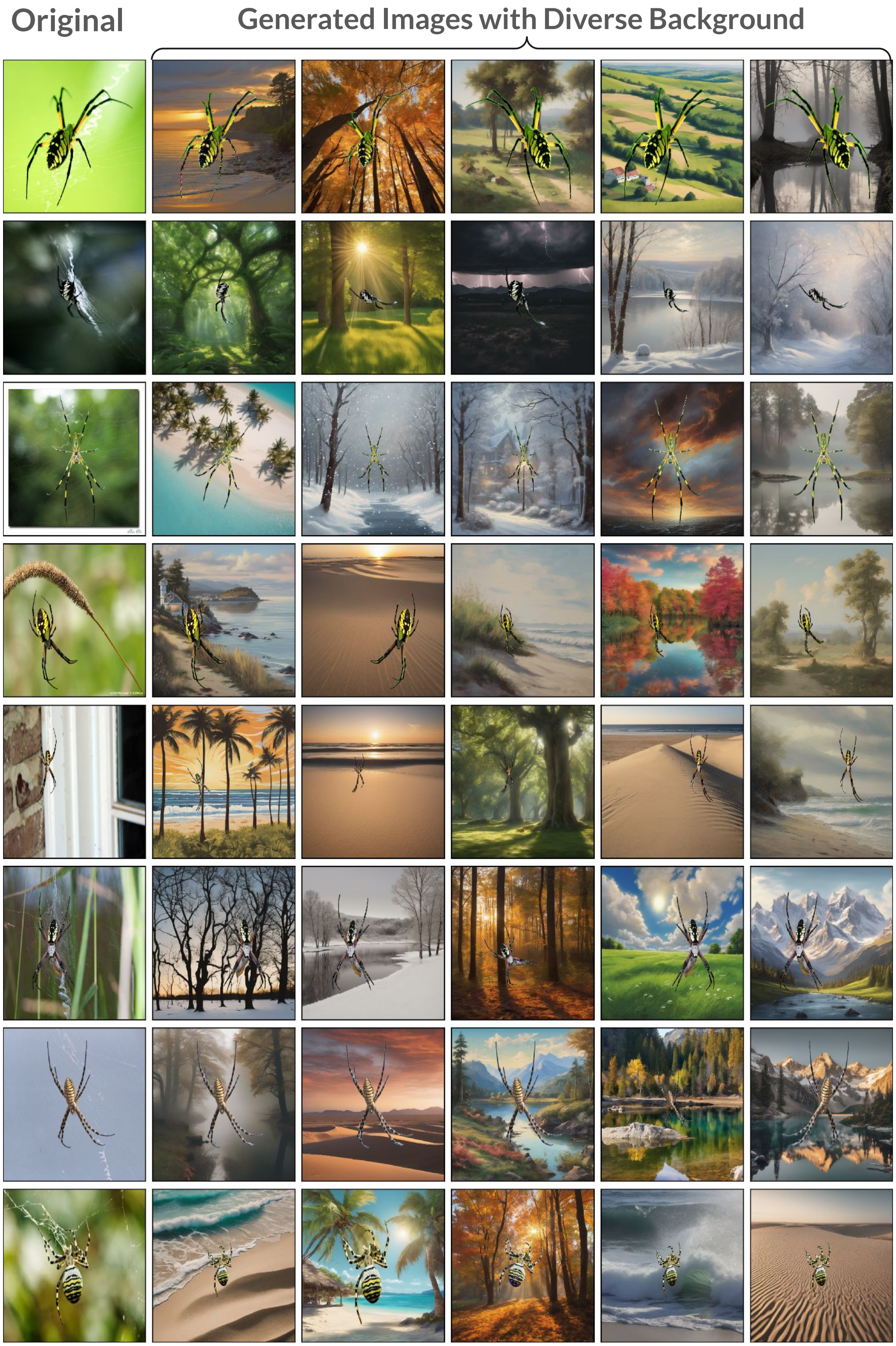


Figure 3. The figure displays the original image samples from ImageNet10 and the generated images using AGA .

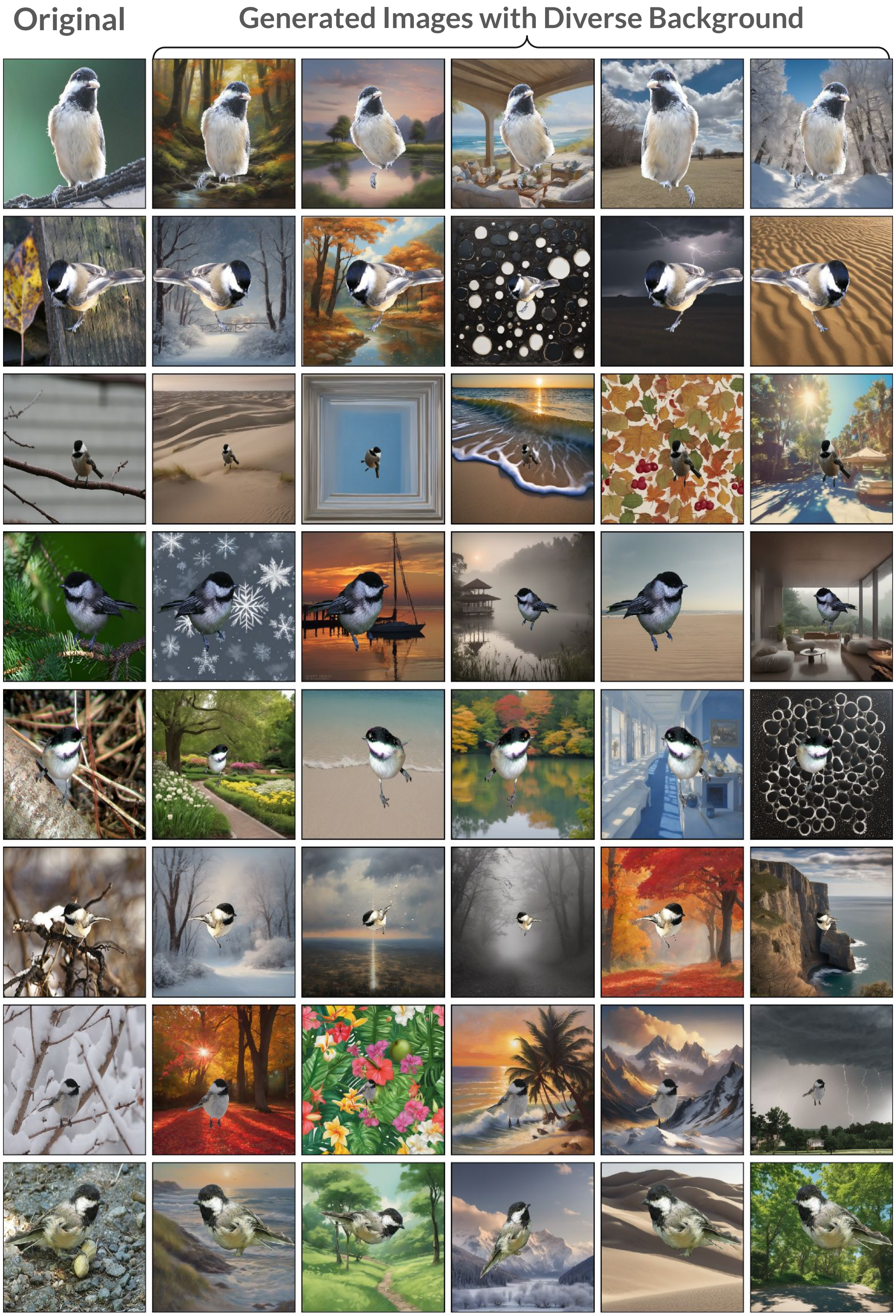


Figure 4. The figure displays the original image samples from ImageNet10 and the generated images using AGA .

3.2. GradCam Visualization Results

We present additional GradCam visualization results here to show the explanable capability of the base model and AGA model for ImageNet 10. The base model classifier is trained with only the real images of the ImageNet10 dataset, but the AGA model is trained with real images as well as augmented images generated by the AGA method.

We demonstrate several validation dataset samples of ImageNet10 that are misclassified by base model in Figure 5. On the other hand, the AGA model correctly classified these data samples, and the following GradCam visualizations reveal that the baseline model often focuses on irrelevant pixels, whereas the AGA-trained model more accurately targets pixels within the subject area.

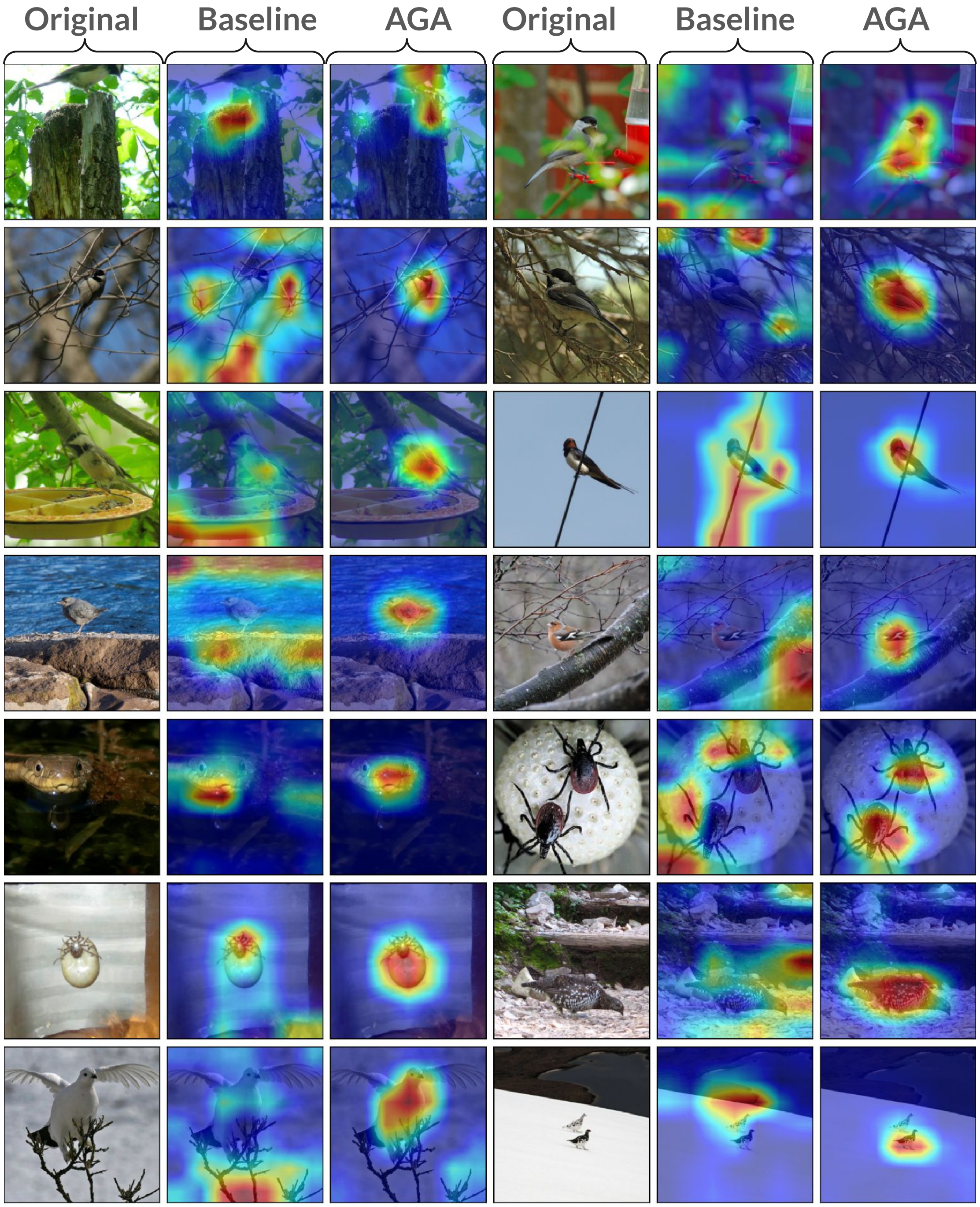
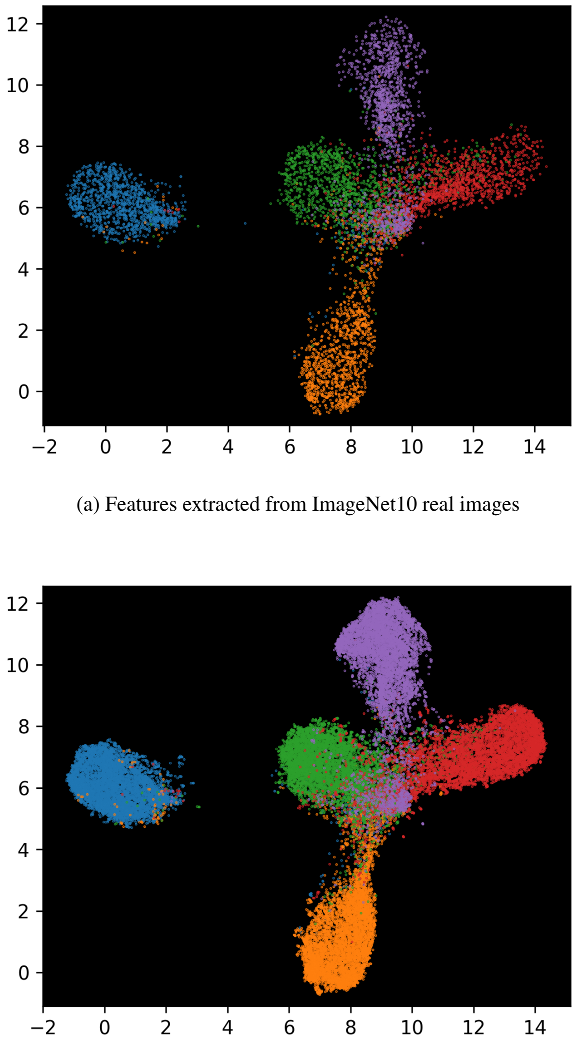


Figure 5. The figure displays additional GradCam visualization results for ImageNet10 dataset.

3.3. Feature Cluster

We conducted an additional experiment to verify that additional synthetic images do not introduce irrelevant features. We utilize the last-layer feature outputs from the ResNet-50 model for both ImageNet10 real and AGA-augmented images. Each image yields 2048 features, which we use to plot feature clusters. We illustrate five distinct class clusters of ImageNet10 for both real and AGA-augmented images in Figure 6. The figure shows that additional generated images enhance cluster density without significantly increasing inter-cluster distances.



(a) Features extracted from ImageNet10 real images

(b) Features extracted from augmented images generated by AGA

Figure 6. UMAP plot of feature clusters of five distinct classes of ImageNet10 dataset where features are extracted from last layer of ResNet-50 model.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[2] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 1

[3] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1

[4] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[5] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[6] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 1

[7] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1

[8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[9] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 1

[10] How to Train State-Of-The-Art Models Using TorchVision's Latest Primitives — pytorch.org. https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/#baseline. [Accessed 24-04-2024]. 1