Appendix of “Low-Biased General Annotated Dataset Generation”

Dengyang Jiang[1*] Haoyu Wang[1*] Lei Zhang[1†] Wei Wei[1]
Guang Dai[2] Mengmeng Wang[3] Jingdong Wang[4] Yanning Zhang[1]

[1] Northwestern Polytechnical University [2] SGIT AI Lab, State Grid Corporation of China
[3] Zhejiang University of Technology [4] Baidu Inc.

1. Loss Computation Algorithm

Algorithm 1 A complete loss computation step for the lbGen generator during fine-tuning

Input: class name c, semantic description p_c, text features of classnames $\{f_{c_1}, \ldots, f_{c_{1000}}\}$, generator ϵ_θ, CLIP model $\mathcal{C}$, discriminator $\mathcal{D}_\phi$, Q-ALIGN model $\mathcal{Q}$, noise ξ, scaler λ_1.

$im = \text{GenerateImage}(\epsilon_\theta, \xi, c)$
$f_{te} = \text{RandomlySelect}(\{f_{c_1}, \ldots, f_{c_{1000}}\})$
$f_{im}, f_{p_c} = \text{GetFeatures}(\mathcal{C}, im, p_c)$
$\mathcal{L}_{en}, \mathcal{L}_{neg} = \text{ComputeEntireLoss}(\mathcal{D}_\phi, f_{im}, f_{te})$
$\mathcal{L}_{in} = \text{ComputeIndividualLoss}(f_{im}, f_{p_c})$
$\mathcal{L}_{bi} = \mathcal{L}_{en} + \mathcal{L}_{in}$
$\mathcal{L}_q, = \text{ComputeQualityLoss}(\mathcal{Q}, im)$
$\mathcal{L} = \mathcal{L}_{bi} + \lambda_1 \mathcal{L}_q$

Output: Training loss for lbGen generator $\mathcal{L}$.

2. Scoring Quality

Q-ALIGN [21] can be recognized as a special version of the multimodal large language model (MLLM). Given an image and system prompt, Q-ALIGN can generate a set of tokens including a <LEVEL> token which represents a probability distribution (denoted as $\mathcal{X}$) over all possible tokens. This distribution is then post-processed to derive a score. In the post-processing phase, a closed-set softmax operation is conducted on the set $\{l_i|_{i=1}^5\}$ = {*bad, poor, fair, good, excellent*} to obtain the probabilities p_{l_i} for each level, such that the sum of p_{l_i} for all l_i equals 1:

$$p_{l_i} = \frac{e^{\mathcal{X}_{l_i}}}{\sum_{j=1}^{5} e^{\mathcal{X}_{l_j}}}. \tag{1}$$

As each text level{*bad, poor, fair, good, excellent*} corresponds to a score{*1, 2, 3, 4, 5*}(higher means better quality), the final predicted score of Q-ALIGN can be formulated as:

$$\mathcal{S}_{\text{II}} = i \times \frac{e^{\mathcal{X}_{l_i}}}{\sum_{j=1}^{5} e^{\mathcal{X}_{l_j}}}, \tag{2}$$

where S_q is ranging from one to five.

3. Training Details

In our fine-tuning method, we inject LoRA layers into the UNet of the diffusion model and train the discriminator from scratch. We keep all other components frozen during training. When training visual backbones, we follow the training recipe in ConvNeXt [13]. It is worth noting that we train Vit-S 40 epochs more than ResNet50 because Transformers often need more time to converge. We provide the detailed training hyperparameters in Table. 4 and Table. 1.

What’s more, when applying the backbones to downstream tasks, we use the toolbox provided in trex [18] to train the linear classifiers for transfer learning. We use MMDetection [3] and MMSegmentation [5] toolboxes to train the detection heads and segmentation heads for visual perception tasks, respectively. In the few-shot [20] setup, we keep the number of training epochs consistent rather than the number of iterations.

4. Data Synthesis Details

We use SD1.5 [17] across all benchmarks. Besides, text prompt “classnames" and hyperparameters showd in Table 3 are used to synthesize ImageNet-like datasets (IN-1k, IN-100).

Model	Sampling steps	Scheduler	Guidance scale	Image size
SD1.5	50	PNDM [12]	2.0	512×512

Table 3. Hyperparameters used when synthesizing data.

5. Datasets Details

Except for ImageNet, We also compare with other two synthetic ImageNet datasets [1, 24] because they are the only

*Equal contribution.
†Corresponding author

Table 1. Training hyperparameters of **visual backbones**.

Name	**ResNet50**	**ViT-S**	**ResNet50**(ablation)
Learning rate	1e-3	1e-3	1e-3
Learning rate scheduler	Cosine decay	Cosine decay	Cosine decay
Epochs	120	160	120
LR warmup epochs	12	16	12
Total batch size	2048	2048	512
Optimizer	AdamW	AdamW	AdamW
AdamW - β_1	0.9	0.9	0.9
AdamW - β_2	0.999	0.999	0.999
RandAugment	(9, 0.5)	(9, 0.5)	(9, 0.5)
Mixup	0.8	0.8	0.8
CutMix	1.0	1.0	1.0
Random erasing	0.25	0.25	0.25
Label smoothing	0.1	0.1	0.1
Stochastic depth	0.1/0.4/0.5/0.5	0.1/0.4/0.5/0.5	0.1/0.4/0.5/0.5
Layer scale	1e-6	1e-6	1e-6
Head init scale	None	None	None
Gradient clip	None	None	None
Exp. Mov. Avg. (EMA)	0.9999	0.9999	0.9999

Dataset	# Classes	# Train samples	# Val samples	# Test samples	Val provided	Test provided
ImageNet validation sets (training classes)						
ImageNet-Val (IN-val) [6]	1000	–	–	50000	–	✓
ImageNet100-Val (IN100-val) [19]	100	–	–	5000	–	✓
Transfer learning(novel classes)						
Aircraft [14]	100	3334	3333	3333	✓	✓
Cars196 [10]	196	5700	2444	8041	–	✓
DTD [4]	47	1880	1880	1880	✓	✓
EuroSAT [8]	10	13500	5400	8100	–	–
Flowers [15]	102	1020	1020	6149	✓	✓
Pets [16]	37	2570	1110	3669	–	✓
Food101 [2]	101	68175	7575	25250	–	✓
Sun397 [22]	397	15880	3970	19850	–	✓
Specific bias (original training classes)						
Cue Conflict [7]	16	–	–	1280	–	✓
FOCUS [9]	226	–	–	23902	–	✓
Mixed-Rand & Mixed-Same [23]	9	–	–	8100	–	✓
Visual perception						
COCO [11]	80	118287	5000	40670	✓	✓
ADE20K [25]	150	20210	2000	3000	✓	✓

Table 2. **Datasets** we use for evaluating the models.

Name	SD1.5
Dataset Generator	
Learning rate	2e-5
Learning rate scheduler	Constant
LR warmup steps	0
Optimizer	AdamW
AdamW - β_1	0.9
AdamW - β_2	0.999
Gradient clipping	0.1
Discriminator	
Learning rate	1e-5
Learning rate scheduler	Constant
Optimizer	AdamW
AdamW - β_1	0
AdamW - β_2	0.999
Gradient clipping	1.0
Quality assurance loss weight λ_2	0.1
Gradient enable steps	5
LoRA rank	128
Classifier-free guidance scale	2
Resolution	512×512
Total training epochs	3
Local batch size	4
Mixed Precision	FP16

Table 4. lbGen training hyperparameters for SD1.5.

Figure 1. Visualization of generated images prompted by polysemy class name in our dataset.

open source datasets based on SD1.5. Thus, we can get fairer and more convincing results based on one implementation. In addition, all datasets used in our metrics to benchmark the bias of the datasets and test the generalization capacities of the backbones are listed in the Table 2.

6. Computing Resources

It takes about 1 hour to fine-tune the generator and 52 hours to generate the ImageNet-like dataset ($\sim$1.3M images) with 8 A100-80G GPUs. The generation runtime of each image is comparable to existing diffusion models.

7. Limitation

While our lbGen demonstrates a great potential to obtain low-biased annotated dataset like ImageNet, the polysemy of some text descriptions may bring drawbacks. As shown in Figure 1, some divergences occur when the class name refers to several objects . For instance, the text "`crane`" can denote either a bird or a machine, and when prompted with "`crane`" to generate a class in our dataset, two entirely different objects will appear. We consider that these divergences are caused by the multiple directions of clip text space due to the polysemy of human words and may compromise the knowledge of classification models trained on our dataset. Although we believe this issue can be solved with more specific text descriptions instead of class names, how to introduce more specific text descriptions without additional bias other than object is still unclear. We will explore it in our future works.

What's more, our method attempts employing the low-biased text information (e.g., object category name) to regularize and fine-tune the diffusion model in the CLIP feature space for low-biased image generation. Although the diffusion model is only fine-tuned on the 1K categories in ImageNet, our generated dataset shows less bias (i.e., better generalization capacity in downstream tasks) than other competitors. However, on one hand, since the fine-grained categories in ImageNet are scarce, the generalization performance of our method in fine-grained object recognition tasks is still limited. On the other hand, compared with the infinite categories of objects in real world, the number of categories employed for fine-tuning remains limited. This also restrict the generalization capacity of our method, i.e., produces bias. Fortunately, our method provides a general low-biased dataset generation framework, which can mitigate both limitations mentioned above by simply introducing more object categories for fine-tuning.

References

[1] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023. 1

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 2

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy

Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 2

[5] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020. 1

[6] 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. 2

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 2

[8] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207. IEEE, 2018. 2

[9] Priyatham Kattakinda and Soheil Feizi. Focus: Familiar objects in common and uncommon settings. In *International Conference on Machine Learning*, pages 10825–10847. PMLR, 2022. 2

[10] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2013. 2

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[12] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. 1

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 1

[14] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2

[15] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2

[16] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 2

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[18] Mert Bulent Sariyildiz, Yannis Kalantidis, Karteek Alahari, and Diane Larlus. No reason for no supervision: Improved generalization in supervised models. *arXiv preprint arXiv:2206.15369*, 2022. 1

[19] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 2

[20] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020. 1

[21] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023. 1

[22] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 2

[23] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020. 2

[24] Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. Real-fake: Effective training data synthesis through distribution matching. *arXiv preprint arXiv:2310.10402*, 2023. 1

[25] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 2