ForAug: Recombining Foregrounds and Backgrounds to Improve Vision Transformer Training with Bias Mitigation – Supplementary Material –

Anonymous submission

Abstract

This is the supplementary material for the paper: *ForAug*: Recombining Foregrounds and Backgrounds to Improve Vision Transformer Training with Bias Mitigation

Extended Bates Distribution

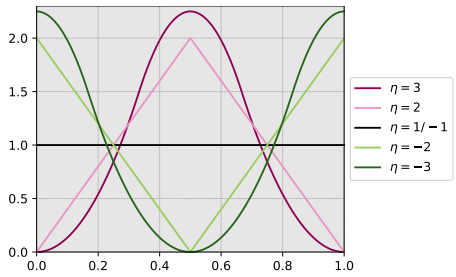


Figure 1: Plot of the probability distribution function (PDF) of the extended Bates distribution for different parameters η. Higher values of η concentrate the distribution around the center.

We introduce an extension of the Bates distribution (Bates 1955) to include negative parameters, enabling sampling of foreground object positions away from the image center. The standard Bates distribution, for $\eta \in \mathbb{N}$, is defined as the mean of η independent random variables drawn from a uniform distribution (Jonhson, Kotz, and Balakrishnan 1995). A larger η value increases the concentration of samples around the distribution's mean, which in this case is the image center.

To achieve an opposite effect–concentrating samples at the image borders–we extend the distribution to $\eta \leq 1$.

$$X \sim \text{Bates}(\eta) :\Leftrightarrow s(X) \sim \text{Bates}(-\eta)$$

This is accomplished by sampling from a standard Bates distribution with parameter $-\eta \geq 1$ and then applying a sawtooth function. The sawtooth function on the interval $[0, 1]$ is defined as

$$s(x) = \begin{cases} x + 0.5 & \text{if } 0 < x < 0.5 \\ x - 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases} \tag{1}$$

This function effectively maps the central portion of the interval to the edges and the edge portions to the center. For example, a value of 0.3 (central-left) is mapped to 0.8 (edge-right), while 0.8 (edge-right) is mapped to 0.3 (central-left). This transformation inverts the distribution's concentration, shifting the probability mass from the center to the borders. We visualize the distribution function of the extended Bates distribution in Figure 1. Both $\eta = 1$ and $\eta = -1$ result in a uniform distribution across the image.

Resource Usage of *ForAug*

To utilize the proposed *ForAug*, specific computational resources are necessary, particularly in terms of storage for the output of the segmentation stage and on-the-fly processing of the recombination stage. The output of *ForAug*/*ForNet*'s segmentation step on ImageNet dataset requires 73 GB of additional disk space for the segmentation output, which is separate from the base 147 GB ImageNet size. The recombination step of *ForAug* is implemented as a based data loader operation. It's thus offloaded to the CPU, where it can be heavily parallelized and thus only results in a very minor increase in the training step-time. For example, using a ViT-B model on an NVIDIA A100 GPU, the average update step-time increased by 1%, from 528 ± 2 ms to 534 ± 1 ms.

Training Setup

Parameter	ViT, Swin, ResNet	DeiT
Image Resolution	224×224	224×224
Epochs	300	300
Learning Rate	3e-3	S/B: 1e-3, L: 5e-4
Learning Rate Schedule	cosine decay	cosine decay
Batch Size	2048	1024
GPUs	$4\times$ NVIDIA A100/H100/H200	$4\times$ NVIDIA A100/H100/H200
Warmup Schedule	linear	linear
Warmup Epochs	3	3
Weight Decay	0.02	0.05
Label Smoothing	0.1	0.1
Optimizer	Lamb (You et al. 2020)	AdamW
Data Augmentation Policy	3-Augment (Touvron, Cord, and Jégou 2022)	DeiT (Touvron et al. 2021)

Table 1: Training setup and hyperparameters for our ImageNet and *ForNet* training.

Dataset	Batch Size	Epochs	Learning Rate	Num. GPUs
Aircraft	512	500	3e-4	2
Cars	1024	500	3e-4	4
Flowers	256	500	3e-4	1
Food	2048	100	3e-4	4
Pets	512	500	3e-4	2

Table 2: Training setup for finetuning on different downstream datasets. Other settings are the same as in Table 1. For finetuning, we always utilize 3-Augment and the related parameters from the *ViT, Swin, ResNet* column of Table 1

On ImageNet we use the same training setup as (Nauen, Palacio, and Dengel 2025) and (Touvron, Cord, and Jégou 2022) without pretraining for ViT, Swin, and ResNet. For DeiT, we train the same ViT architecture but using the data augmentation scheme and hyperparameters from (Touvron et al. 2021). As our focus is on evaluating the changes in accuracy due to *ForAug*/*ForNet*, like (Nauen, Palacio, and Dengel 2025), we stick to one set of hyperparameters for all models. We list the settings used for training on ImageNet and *ForNet* in Table 1 and the ones used for finetuning those weights on the downstream datasets in Table 2. Out implementation is using PyTorch (Paszke et al. 2019) and the *timm* library (Wightman 2019) for model architectures and basic functions.

Parameter	Value
GPU	NVIDIA A100/H100/H200
CPU	24 CPU cores (Intex Xenon) per GPU
Memory	up to 120GB per GPU
Operating System	Enroot container for SLURM based on Ubuntu 24.04 LTS
Python	3.12.3
PyTorch	2.7.0
TorchVision	0.22.0
Timm	1.0.15

Table 3: Hardware and Software specifics used for both training and evaluation.

Table 3 lists the specific hardware we use, as well as versions of the relevant software packages.

Infill Model Comparison

Table 4: Example infills of LaMa and Attentive Eraser.

We visualize example infilled images for both LaMa (Suvorov et al. 2021) and Attentive Eraser (Sun et al. 2024) in Table 4. We qualitatively find that while LaMa often leaves repeated textures of blurry spots where the object was erased, Attentive Eraser produces slightly cleaner and more coherent infills of the background.

Images with High Infill Ratio

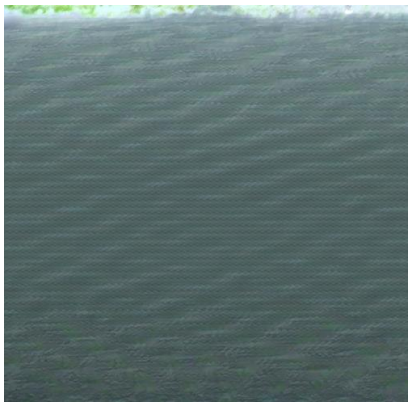
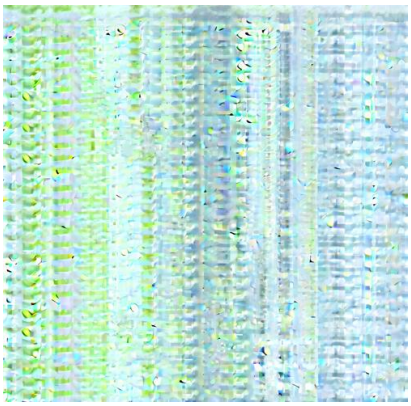
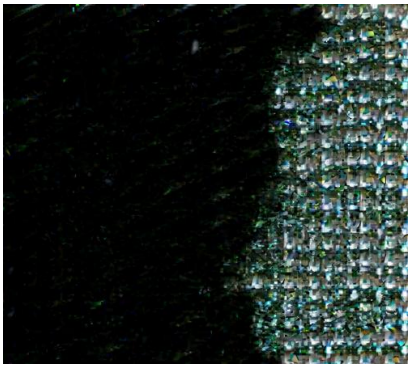
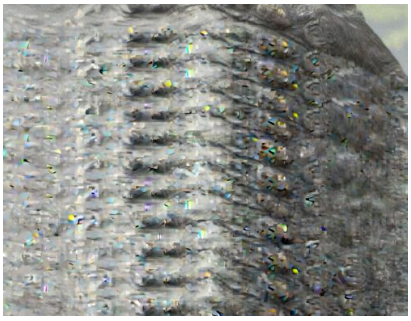
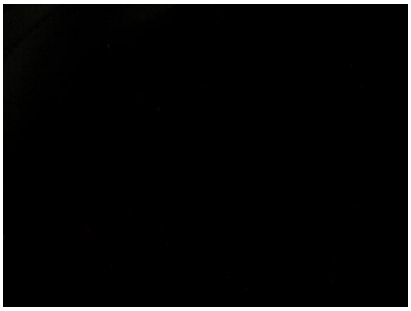
Infill Ratio	LaMa	Att. Eraser
93.7		
95.7		
83.7		
88.2		

Table 5: Example infills with a large relative foreground area size that is infilled (infill ratio).

Table 5 shows infills for images where Grounded SAM (Ren et al. 2024) marks a high percentile of the image as the foreground object (Infill Ratio), that has to be erased by the infill models. While LaMa tends to fill those spots with mostly black or gray

and textures similar to what we saw in Table 4, Attentive Eraser tends to create novel patterns by copying what is left of the background all over the rest of the image. We filter out all backgrounds that have an infill ratio larger than our pruning threshold $t_{\text{prune}} = 0.8$.

References

Bates, G. 1955. Joint distributions of time intervals for the occurrence of successive accidents in a generalized Polya urn scheme. *Annals of Mathematical Statistics*, 26: 705–720.

Jonhson, N. L.; Kotz, S.; and Balakrishnan, N. 1995. *Continuous Univariate Distributions*. Wiley series in probability and mathematical statistics. Wiley, 2 edition. ISBN 0-471-58494-0. Wiley series in probability and mathematical statistics.

Nauen, T. C.; Palacio, S.; and Dengel, A. 2025. Which Transformer to Favor: A Comparative Analysis of Efficiency in Vision Transformers. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 6955–6966.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Ren, T.; Liu, S.; Zeng, A.; Lin, J.; Li, K.; Cao, H.; Chen, J.; Huang, X.; Chen, Y.; Yan, F.; Zeng, Z.; Zhang, H.; Li, F.; Yang, J.; Li, H.; Jiang, Q.; and Zhang, L. 2024. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks.

Sun, W.; Cui, B.; Dong, X.-M.; and Tang, J. 2024. Attentive Eraser: Unleashing Diffusion Model's Object Removal Potential via Self-Attention Redirection Guidance.

Suvorov, R.; Logacheva, E.; Mashikhin, A.; Remizova, A.; Ashukha, A.; Silvestrov, A.; Kong, N.; Goka, H.; Park, K.; and Lempitsky, V. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jegou, H. 2021. Training data-efficient image transformers & distillation through attention. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 10347–10357. PMLR.

Touvron, H.; Cord, M.; and Jégou, H. 2022. DeiT III: Revenge of the ViT. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 516–533. Cham: Springer Nature Switzerland.

Wightman, R. 2019. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models.

You, Y.; Li, J.; Reddi, S.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; and Hsieh, C.-J. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*.