CNS-Bench: Benchmarking Image Classifier Robustness Under Continuous Nuisance Shifts

Supplementary Material

A. Benchmark Details

This section provides more details about the benchmarking dataset.

A.1. List of Shifts, Classes, and Example Images

The results are averaged over the following 14 shifts: *cartoon style, plush toy style, pencil sketch style, painting style, design of sculpture, graffiti style, video game renditions style, style of a tattoo, heavy snow, heavy rain, heavy fog, heavy smog, heavy dust, heavy sandstorm* (see examples in Fig. 34 and Fig. 35). We train the sliders using the prompt template "`A picture of a {class} in {shift}`". Here, we consider the following classes: *hammerhead, hen, ostrich, junco, bald eagle, common newt, tree frog, african chameleon, scorpion, centipede, peacock, toucan, goose, koala, jellyfish, hermit crab, pelican, sea lion, afghan hound, bloodhound, italian greyhound, whippet, weimaraner, golden retriever, collie, border collie, rottweiler, french bulldog, s aint bernard, siberian husky, dalmatian, pug, pembroke, red fox, leopard, snow leopard, lion, ladybug, ant, mantis, starfish, wood rabbit, fox squirrel, beaver, hog, hippopotamus, bison, skunk, gibbon, baboon, giant panda, eel, puffer, accordion, ambulance, basketball, binoculars, birdhouse, bow tie, broom, bucket, cannon, canoe, carousel, cowboy hat, fire engine, flute, gasmask, grand piano, hammer, harp, hatchet, jeep, joystick, lipstick, mailbox, mitten, parachute, pickup, sax, school bus, soccer ball, submarine, tennis ball, warplane, ice cream, bagel, pretzel, cheeseburger, hotdog, head cabbage, broccoli, cucumber, bell pepper, granny smith, lemon, burrito, espresso, volcano, ballplayer*.

B. More Benchmarking Results

Fig. 9 presents the accuracy drops averaged over all shifts and Tab. 5 lists all average accuracies and accuracy drops for all evaluated models and shift scales. Fig. 11 plots the accuracy drops for painting, cartoon, and snow shifts with confidence intervals. As discussed in the main paper, we also provide the accuracy drops for the ResNet family in Fig. 12. Similar to the observations in Tab. 3, larger models result in a lower accuracy drop in average. Fig. 10 provides a more nuanced view on the model performances across various architectures on all shifts. We also plot failure point distributions in Fig. 13. Fig. 15 presents more classifier results on the labeled dataset.

The accuracies for the diffusion classifier are depicted in Fig. 14. Similar to the discussion in the paper, the results showcase that the generative classifier is less robust than a supervised classifier. We use the DiT-based diffusion classifier trained on ImageNet-1k using the available framework [33] and the default hyper-parameters with a resolution of 256. Due to high computational costs, we compute the results for 100 classes, four scales, for the snow and cartoon style shift, and for at most 20 seeds per class, scale, and shift.

C. Fine-tuning with Synthetic Data

We fine-tune a ResNet-50 classifier using our synthetic data. We compare the original ImageNet-trained model to a model fine-tuned using 50% synthetic data and 50% ImageNet training data. As shown in Tab. 6, the fine-tuned model leads to improved performance on the shifted real-world dataset, without a significant decline on the original ImageNet dataset.

D. Accuracy Drops on ImageNet-C

We provide more evidence that the model rankings can change for different scales for ImageNet-C as well. We consider seven levels of contrast as a deterministic example corruption from ImageNet-C, based on the implementation of Hendrycks and Dietterich [24]. We present the accuracy drops for all corruption levels in Fig. 16 and Fig. 17. Similar to our benchmark, a global averaged metric fails to capture such variations.

E. Comparison to ImageNet-C/R

While ImageNet-R evaluates style shifts, it includes confounders, such as heavy shape and perspective changes (Fig. 19). Our approach aims at reducing such factors by reducing variations of the spatial structure of the image when gradually applying the shift.

F. Discussion of Accuracy-on-the-Line

We observe that larger models obtain higher OOD accuracies, *i.e.*, smaller accuracy drops, as shown in Fig. 9 (*Model size*). However, ID and OOD accuracy are correlated, as we show in Fig. 22. As prior work [43] has shown that ID and OOD accuracy relate linearly, *i.e.*, *accuracy on the line*, we want to study whether the larger parameter count

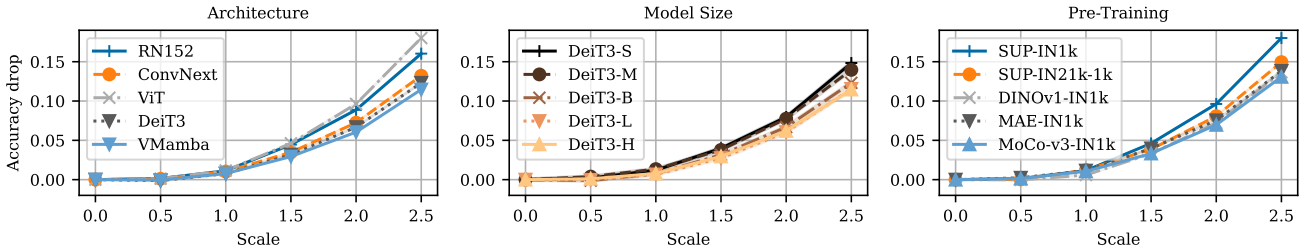


Figure 9. **Accuracy drops averaged over the whole benchmark.** Architecture (*left*): We show models with the same training data and similar parameter counts. The selection of the architecture influences the accuracy drop. Model size (*center*): We show DeiT3 with various numbers of parameters. Increasing the model capacity results in lower accuracy drops. Pre-training paradigm and data (*right*): We show different pre-training paradigms: supervised, self-supervised (MAE, DINO, MoCo), and more data (IN21k), all using ViT-B/16. We present results for all shifts in Fig. 10.

Table 4. **mCE and mean rCE.** We present the mean corruption error and the mean relative corruption error for all evaluated models.

	CE	rCE
alexnet	1.000	1.000
clip_resnet101	0.532	0.563
clip_resnet50	0.715	0.587
clip_vit_base_patch16_224	0.420	0.230
clip_vit_base_patch32_224	0.487	0.591
clip_vit_large_patch14_224	0.445	0.228
clip_vit_large_patch14_336	0.419	0.274
convnext_base.fb_in1k	0.359	0.686
convnext_large.fb_in1k	0.354	0.672
convnext_small.fb_in1k	0.353	0.609
convnext_tiny.fb_in1k	0.393	0.809
convnextv2_base.fcmae_ft_in1k	0.322	0.680
convnextv2_huge.fcmae_ft_in1k	0.283	0.553
convnextv2_large.fcmae_ft_in1k	0.297	0.568
deit3_base_patch16_224.fb_in1k	0.396	0.610
deit3_huge_patch14_224.fb_in1k	0.353	0.583
deit3_large_patch16_224.fb_in1k	0.382	0.574
deit3_medium_patch16_224.fb_in1k	0.387	0.758
deit3_small_patch16_224.fb_in1k	0.400	0.747
deit_base_patch16_224.fb_in1k	0.437	0.746
dino_vit_base_patch16	0.504	0.851
dinov1_vit_base_patch16	0.412	0.676
dinov2_vit_base_patch14	0.350	0.524
dinov2_vit_base_patch14_reg	0.311	0.456
dinov2_vit_giant_patch14	0.321	0.431
dinov2_vit_giant_patch14_reg	0.311	0.426
dinov2_vit_large_patch14	0.298	0.349
dinov2_vit_large_patch14_reg	0.296	0.370
dinov2_vit_small_patch14	0.351	0.639
dinov2_vit_small_patch14_reg	0.330	0.627
mae_vit_base_patch16	0.386	0.732
mae_vit_huge_patch14	0.303	0.542
mae_vit_large_patch16	0.328	0.571
mocov3_vit_base_patch16	0.379	0.669
resnet101.a1_in1k	0.491	0.842
resnet152.a1_in1k	0.498	0.790
resnet18.a1_in1k	0.493	0.954
resnet34.a1_in1k	0.440	0.843
resnet50.a1_in1k	0.485	0.945
vit_base_patch16_224.augreg_in1k	0.569	0.926
vit_base_patch16_224.augreg_in21k_ft_in1k	0.460	0.722
vit_base_patch16_clip_224.openai_ft_in1k	0.282	0.482
vssm_base_v0	0.371	0.574

explains the higher robustness or whether this is solely explained by the accuracy-on-the-line observation. Therefore, we remove the effect of the ID accuracy on the OOD accuracy by computing the partial correlation between model size and OOD accuracy. Fig. 20 show the slopes for various shifts and Fig. 21 provides the p-values of the linear regression corresponding to the presented results in Fig. 20. This partial correlation coefficient is significantly negative ($\rho_{\text{size,OOD·ID}}$ = -0.358 for the DeiT3 family). Therefore, we conclude from our analysis that the improvements can be explained by the improved ID accuracy but not by more parameters.

We further explore how removing the linear relation (as, e.g., in Fig. 23) explains the better OOD accuracy in Fig. 24.

Table 5. **Accuracy evaluations.** We present the accuracies and accuracy drops of all evaluated classifiers.

	Shift Scale											
	Accuracy							Accuracy Drop				
model	0	0.5	1	1.5	2	2.5	avg	1	1.5	2	2.5	avg
clip_resnet50	0.81	0.81	0.8	0.78	0.74	0.67	0.77	0.01	0.03	0.07	0.14	0.04
clip_resnet101	0.86	0.86	0.85	0.83	0.81	0.74	0.82	0.01	0.03	0.06	0.12	0.04
clip_vit_base_patch16_224	0.87	0.88	0.88	0.87	0.86	0.81	0.86	-0.00	0.01	0.02	0.06	0.02
clip_vit_base_patch32_224	0.87	0.87	0.86	0.85	0.83	0.77	0.84	0.01	0.02	0.04	0.1	0.03
clip_vit_large_patch14_224	0.87	0.87	0.87	0.86	0.85	0.82	0.86	-0.00	0.01	0.02	0.05	0.01
clip_vit_large_patch14_336	0.88	0.88	0.88	0.87	0.86	0.83	0.87	0.00	0.01	0.02	0.05	0.01
convnext_tiny.fb_in1k	0.92	0.92	0.91	0.88	0.84	0.77	0.87	0.01	0.04	0.08	0.15	0.05
convnext_small.fb_in1k	0.92	0.93	0.92	0.89	0.86	0.8	0.89	0.01	0.03	0.07	0.13	0.04
convnext_base.fb_in1k	0.93	0.93	0.92	0.89	0.85	0.79	0.89	0.01	0.03	0.07	0.13	0.04
convnext_large.fb_in1k	0.93	0.92	0.92	0.89	0.86	0.8	0.89	0.01	0.04	0.07	0.12	0.04
convnextv2_base.fcmae_ft_in1k	0.93	0.93	0.92	0.9	0.87	0.82	0.9	0.01	0.04	0.07	0.12	0.04
convnextv2_large.fcmae_ft_in1k	0.94	0.93	0.93	0.91	0.88	0.84	0.91	0.01	0.03	0.05	0.1	0.03
convnextv2_huge.fcmae_ft_in1k	0.94	0.93	0.93	0.91	0.89	0.84	0.91	0.01	0.03	0.05	0.09	0.03
deit3_small_patch16_224.fb_in1k	0.92	0.92	0.91	0.88	0.84	0.77	0.87	0.01	0.04	0.08	0.15	0.05
deit3_base_patch16_224.fb_in1k	0.91	0.91	0.9	0.88	0.84	0.79	0.87	0.01	0.03	0.07	0.12	0.04
deit3_medium_patch16_224.fb_in1k	0.92	0.92	0.91	0.88	0.84	0.78	0.88	0.01	0.04	0.08	0.14	0.05
deit3_large_patch16_224.fb_in1k	0.91	0.91	0.9	0.88	0.85	0.8	0.88	0.01	0.03	0.06	0.12	0.04
deit3_huge_patch14_224.fb_in1k	0.92	0.92	0.91	0.89	0.86	0.81	0.89	0.01	0.03	0.06	0.11	0.04
deit_base_patch16_224.fb_in1k	0.9	0.9	0.89	0.87	0.83	0.76	0.86	0.01	0.04	0.08	0.15	0.05
dino_lp_vit_base_patch16	0.9	0.9	0.89	0.85	0.8	0.71	0.84	0.01	0.05	0.1	0.19	0.06
dinov1_ft_vit_base_patch16	0.91	0.91	0.90	0.88	0.84	0.84	0.87	0.01	0.03	0.07	0.04	0.03
dinov2_vit_small_patch14	0.92	0.92	0.91	0.89	0.86	0.81	0.89	0.01	0.03	0.06	0.11	0.04
dinov2_vit_small_patch14_reg	0.93	0.93	0.92	0.9	0.87	0.81	0.89	0.01	0.03	0.06	0.11	0.04
dinov2_vit_base_patch14	0.91	0.91	0.91	0.89	0.87	0.82	0.89	0.00	0.02	0.04	0.09	0.02
dinov2_vit_base_patch14_reg	0.92	0.92	0.92	0.9	0.88	0.84	0.9	0.00	0.02	0.04	0.08	0.02
dinov2_vit_large_patch14	0.92	0.92	0.92	0.91	0.89	0.86	0.9	0.00	0.01	0.03	0.06	0.02
dinov2_vit_large_patch14_reg	0.92	0.92	0.91	0.91	0.89	0.86	0.9	0.00	0.01	0.03	0.06	0.02
dinov2_vit_giant_patch14	0.91	0.91	0.91	0.9	0.88	0.84	0.89	0.00	0.01	0.04	0.07	0.02
dinov2_vit_giant_patch14_reg	0.92	0.92	0.91	0.9	0.88	0.85	0.9	0.00	0.01	0.03	0.07	0.02
mae_vit_base_patch16	0.92	0.92	0.91	0.88	0.84	0.78	0.88	0.01	0.04	0.08	0.14	0.05
mae_vit_huge_patch14	0.93	0.93	0.92	0.9	0.88	0.84	0.9	0.01	0.03	0.05	0.1	0.03
mae_vit_large_patch16	0.93	0.92	0.92	0.9	0.87	0.83	0.9	0.01	0.03	0.05	0.1	0.03
mocov3_vit_base_patch16	0.92	0.92	0.91	0.88	0.85	0.79	0.88	0.01	0.03	0.07	0.13	0.04
resnet18.a1_in1k	0.9	0.9	0.88	0.85	0.8	0.72	0.84	0.02	0.05	0.1	0.19	0.06
resnet34.a1_in1k	0.91	0.91	0.9	0.86	0.82	0.75	0.86	0.01	0.05	0.09	0.17	0.05
resnet50.a1_in1k	0.91	0.9	0.89	0.85	0.8	0.72	0.85	0.02	0.06	0.11	0.18	0.06
resnet101.a1_in1k	0.9	0.9	0.88	0.85	0.8	0.73	0.84	0.02	0.05	0.1	0.17	0.06
resnet152.a1_in1k	0.89	0.89	0.88	0.85	0.8	0.73	0.84	0.01	0.04	0.09	0.16	0.05
vit_base_patch16_224.augreg_in1k	0.87	0.87	0.86	0.82	0.77	0.69	0.81	0.01	0.05	0.1	0.18	0.06
vit_base_patch16_224.augreg_in21k_ft_in1k	0.9	0.9	0.89	0.86	0.82	0.75	0.85	0.01	0.04	0.08	0.15	0.05
vit_base_patch16_clip_224.openai_ft_in1k	0.93	0.93	0.92	0.91	0.89	0.86	0.91	0.01	0.02	0.04	0.08	0.03
vssm_base_v0	0.91	0.91	0.91	0.89	0.85	0.80	0.88	0.01	0.03	0.06	0.11	0.04

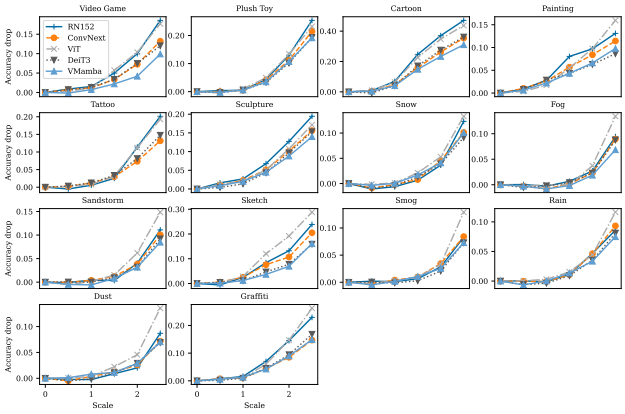


Figure 10. **Accuracy drops of various architectures for all shifts.** We present the accuracy drops for all shifts in our benchmark. The performance gaps vary for different shifts and scales.

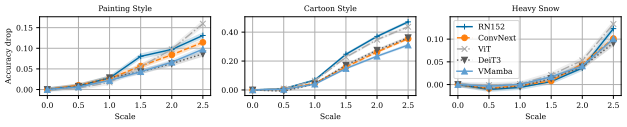


Figure 11. **Accuracy drops with confidence intervals.** The accuracy drops are depicted for the three shifts along the model axes including the one-sigma confidence interval of the accuracy computation. The results show that some ranking changes are statistically stable.

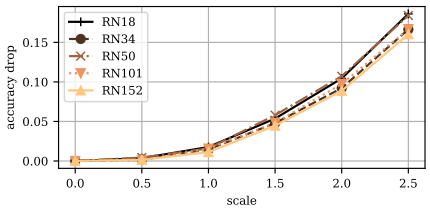


Figure 12. **Robustness evaluation for ResNet model family.** We compute the accuracy drops for all scales when varying the model size for a set of ResNet models. Larger models result in a better OOD robustness.

G. Implementation Details

In this section, we provide more implementation details about the dataset generation process.

G.1. Implementation Details for Image Generation

We use the standard diffusers [61] pipeline for Stable Diffusion 2.0, the DDIM [54] sampler with 100 steps and a guidance scale of 7.5, seeds ranging from 1 to 50.

G.2. Implementation Details for Benchmarking

We integrate our new benchmark and additional models in the easyrobust [40] framework.

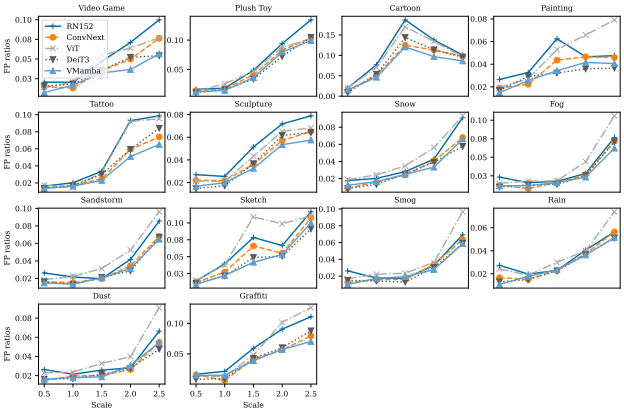
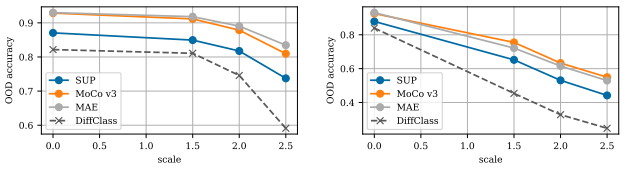


Figure 13. **Failure point distributions for all shifts.** We present the failure point distributions for all shifts in our benchmark. The failure point distributions vary for different shifts, quantifying the different ways the shifts influence model performance.



(a) Accuracies for heavy snow. (b) Accuracies for cartoon style.

Figure 14. **Comparison of DiT classifier.** We report the OOD accuracies for two shifts for the DiT classifier [33] and discriminative classifiers. All models were trained on ImageNet-1k and are evaluated on the same subset of our benchmark. The diffusion classifier performs worse than the discriminative models.

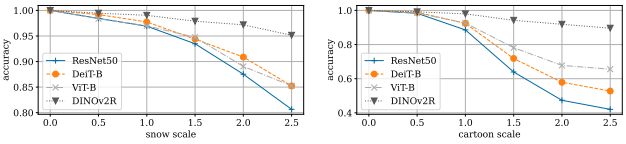


Figure 15. **Classification accuracy on the labeled dataset for snow and cartoon shifts.** The accuracy drops on the labeled dataset showcase that various classifiers have varying sensitivities on different shifts.

Table 6. **ImageNet-R performance after fine-tuning on our benchmark data.**ImageNet-R accuracy of the original ResNet-50 without fine-tuning and our model, fine-tuned on our benchmark.

Evaluation Dataset	wo/ fine-tuning	w/ fine-tuning
IN/val	80.15	78.11
IN/R	27.34	37.57

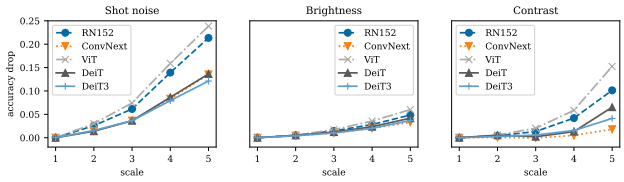


Figure 16. **Accuracy drops for three ImageNet-C corruptions and various architectures.** The model rankings change for different corruptions, underlining the importance of the selection of the corruption types or nuisance shifts for benchmarking the OOD robustness. Additionally, it can also be observed that the accuracy drops at varying rates for different shifts.

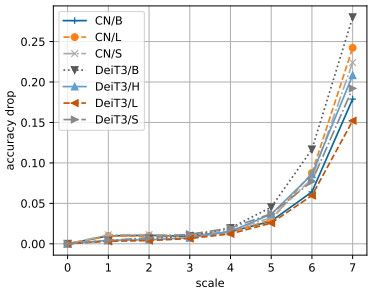


Figure 17. **Accuracy drops for contrast corruption.** We report the accuracy drops for seven severities of the contrast corruption, as defined in [24]. The model rankings change for different scales.

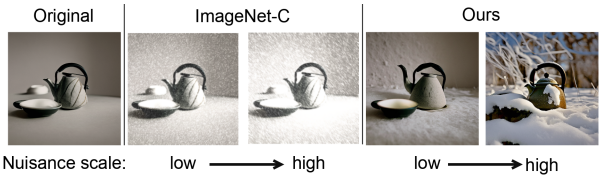


Figure 18. **Illustration of difference between ImageNet-C and CNS.** While ImageNet-C analyzes only synthetic shifts, CNS capture real-world distribution shifts..

Figure 19. **ImageNet-R examples.** Example images of one class where the shape and perspective significantly change.

G.3. Details about the Used Compute

We used the internal cluster consisting of NVIDIA A40, A100, and RTX 8000 GPUs for running most of the experiments. Small-scale experiments are conducted on workstations equipped with RTX 3090. Training one LoRA adapter requires 1 to 2 hours depending on the used GPU, gen-

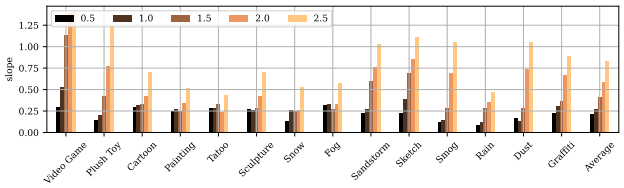


Figure 20. **Slope of ID and OOD accuracies.** We report the slope computed for 16 ImageNet-trained models.

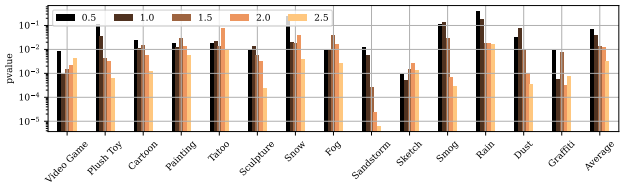


Figure 21. p-values of the linear regressions corresponding to the plot in Fig. 20: The p-value is smaller than 0.05 for most scales and shifts, providing evidence for the statistical significance of our statements.

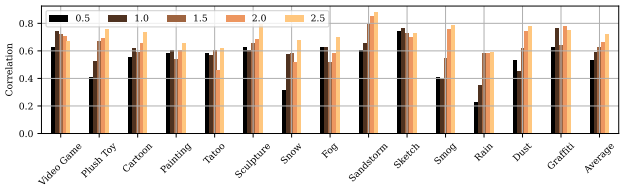
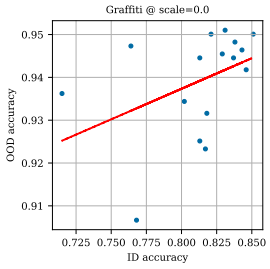
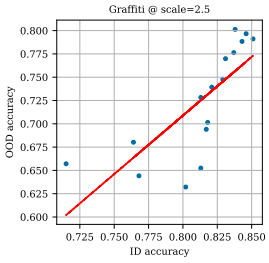


Figure 22. We report the linear correlation coefficients between ID and OOD accuracy using 16 supervised ImageNet-trained models for all evaluated shifts. The relation varies for different shifts and scales between 0.5 and 2.5.



(a) Fit for no applied shift.



(b) Fit for a graffiti shift with scale 2.5.

Figure 23. **Linear fits of the ID and OOD accuracies.** We plot example linear fits of ID and OOD accuracies for the graffiti style. It can be observed that the slope increases for a larger scale.

erating the images for one shift and class with 50 seeds and 6 scales requires 10 to 20 minutes. Thus, the training of the 1400 LoRA adapters took around 2000 GPU hours and the generation of the images around 350 GPU hours.

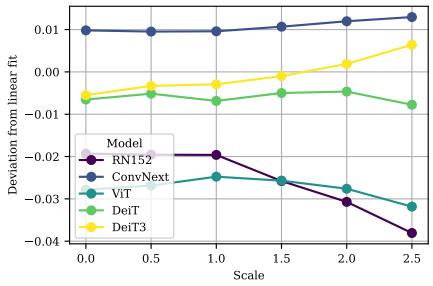


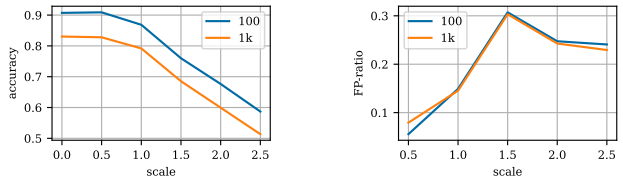
Figure 24. **Accuracy gains of models along the architecture axis.** We plot the accuracy gains averaged over all shifts after correcting for the effect of the ID-OOD accuracy slope. These gains are computed by substracting the effect of the linear fit (consider Fig. 23 for an example) from the OOD accuracies. After that correction, ConvNext performs better than DeiT3.

Table 7. **ImageNet validation accuracies and parameter count**. One the left, we plot model accuracies on the ImageNet validation dataset for all evaluated classifiers. On the right, we present the parameter counts for the used architectures.

Model	IN/val
clip_resnet101	58.00
clip_resnet50	55.00
clip_vit_base_patch16_224	67.70
clip_vit_base_patch32_224	62.60
clip_vit_large_patch14_224	75.00
clip_vit_large_patch14_336	76.30
convnext_base.fb_in1k	83.80
convnext_large.fb_in1k	84.30
convnext_small.fb_in1k	83.10
convnext_tiny.fb_in1k	82.10
convnextv2_base.fcmae_ft_in1k	84.90
convnextv2_huge.fcmae_ft_in1k	86.20
convnextv2_large.fcmae_ft_in1k	85.80
deit3_base_patch16_224.fb_in1k	83.70
deit3_huge_patch14_224.fb_in1k	85.10
deit3_large_patch16_224.fb_in1k	84.60
deit3_medium_patch16_224.fb_in1k	82.90
deit3_small_patch16_224.fb_in1k	81.30
deit_base_patch16_224.fb_in1k	81.80
dino_lp_vit_base_patch16	78.10
dino_v1_vit_base_patch16	82.49
dinov2_vit_base_patch14	84.50
dinov2_vit_base_patch14_reg	84.60
dinov2_vit_giant_patch14	86.60
dinov2_vit_giant_patch14_reg	87.10
dinov2_vit_large_patch14	86.40
dinov2_vit_large_patch14_reg	86.70
dinov2_vit_small_patch14	81.40
dinov2_vit_small_patch14_reg	80.90
mae_vit_base_patch16	83.70
mae_vit_huge_patch14	86.90
mae_vit_large_patch16	86.00
mocov3_vit_base_patch16	83.20
resnet101.a1_in1k	81.30
resnet152.a1_in1k	81.70
resnet18.a1_in1k	71.50
resnet34.a1_in1k	76.40
resnet50.a1_in1k	80.20
vit_base_patch16_224.augreg_in1k	76.80
vit_base_patch16_224.augreg_in21k_ft_in1k	77.70
vit_base_patch16_clip_224.openai_ft_in1k	85.20

Model	Number of parameters (in M)
convnext_tiny	29
convnext_small	50
convnext_base	89
convnext_large	198
convnextv2_base	89
convnextv2_huge	660
convnextv2_large	198
deit3_small	22
deit3_medium	39
deit3_base	87
deit3_huge	632
deit3_large	304
deit_base	87
vit_base	87
vit_huge	632
vit_large	307
resnet18	12
resnet34	22
resnet50	26
resnet101	45
resnet152	60

Benchmarking all models using *easyrobust* required around 1000 GPU hours. The experiments to perform classification using the diffusion-classifier required around 4000 GPU hours.



(a) Accuracy over various scales. (b) Failure point distribution.

Figure 25. **Ablation of the number of ImageNet classes.** We compare the accuracies and failure points averaged over the selected 100 classes and all 1000 ImageNet classes for two shifts (snow and cartoon style). We report the results with ResNet-50. The results indicate that the initial accuracy estimate is overestimated but the accuracy drops averaged over the two shifts are in line. The failure point distribution is normalized.)

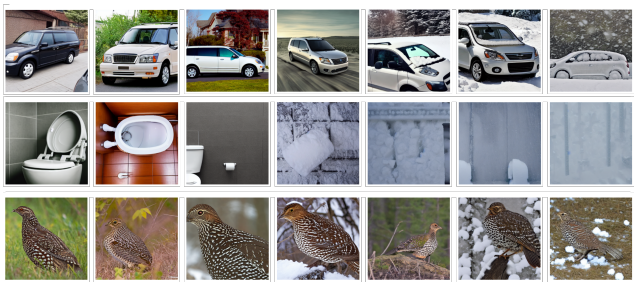


Figure 26. **Examples for text-based continuous shift.** The gradual increase can be successful. However, we observe that it fails for some classes (middle row) and is not always consistently increasing (bottom row).

G.4. Effect of Reduced Number of Classes for Benchmark Evaluation

We ablate how the number of classes influences the robustness evaluations in Fig. 25. For a more efficient computation, we use the `UniPCMultistepScheduler` sampler with 20 steps [72].

H. Design Choice for Text-Based Continuous Shift

A naive approach for realizing continuous shifts involves computing the difference between two corresponding CLIP embeddings. We explored this strategy following the implementation of Baumann et al. [3], but we did not achieve robust nuisance shifts for the variety of classes we considered and we present some examples in Fig. 26. We achieve reasonable results for some classes (*e.g.*, upper row). However, we observed that the spatial structures sometimes changes despite starting at later timesteps. We observed that the naive approach is not very stable for some classes, resulting in OOD samples that do not represent realistic images (*e.g.*, middle row). Applying the delta in text-embedding space also does not always result in a consistent increase of the considered shift (*e.g.*, lower row).

Table 8. **Statistics of filtering process.** We report the number of in-class samples after various filtering stages.

Scale	Stage (i)	Stage (ii)	Stage (iii)	Stage (iv)
0	4000	2966	2966	2966
0.5	4000	2966	2929	2955
1	4000	2966	2813	2906
1.5	4000	2966	2479	2740
2	4000	2966	2143	2498
2.5	4000	2966	1729	2110

We evaluate whether our sliders always increase the shift, as measured by the Δ CLIP score. For this purpose, we compute the Δ CLIP scores when increasing the slider scale by 0.5. Here, the CLIP shift alignment increases for 73% of all cases for scales $s > 0$ and averaged over all shifts, demonstrating that increasing the slider weight results in a stronger severity of the desired shift.

I. Labeling

In this section, we provide more details about the labeling dataset and strategy.

I.1. Details on the Creation of the Labeled Dataset

To select a filter for detecting out-of-class (OOC) samples, we collected a manually labeled dataset. For this, we pursued the following strategy: (i) In the first stage, 24k images are generated for 20 seeds, 5 LoRA scales, and 2 shifts per class for 100 random ImageNet classes in total. We select two different shifts: One shift corresponds to a natural variation (snow), and the second shift corresponds to a style shift (cartoon style). (ii) We aim to find OOC samples that are due to the application of the LoRA adapters. Therefore, we remove all images generated with a seed that results in a generated image with low CLIP text-alignment or that is not classified classified correctly even without the application of LoRA adapters. After removing such images, the labeling dataset consists of around 18k images. (iii) To reduce the labeling effort, we filter out all easy samples that (1) are correctly classified by DINOv2-ViT-L [5, 45] with a linear fine-tuned head and (2) one out of three classifiers (ResNet-50, DeiT-B/16, or ViT-B/16). (3) Additionally, we ensure a sufficiently high text alignment. (iv) The remaining hard images are labeled by two human annotators.

Eeach annotator can choose from the labels ‘class’, ‘partial class properties’, and ‘not class’, where the second option should be selected if the image partially includes some characteristics of the class. An image is defined as an out-of-class sample if at least one annotator considers the image as an OOC sample. For the remaining samples, an image is

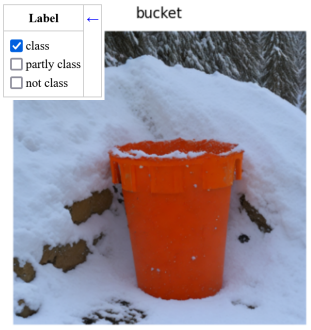
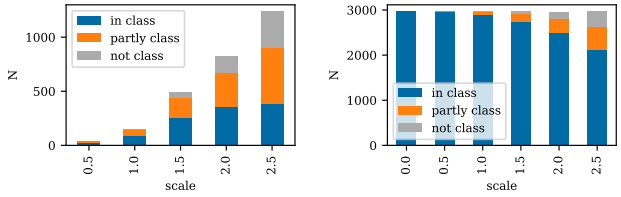


Figure 27. **Screenshot of labeling tool.** We plot a screenshot of an example image as it appeared during our labeling.



(a) Human labeling dataset. (b) Complete filtering dataset.

Figure 28. **Statistics of labeling dataset.** We report the number of in-class, partially in-class, and out-of-class samples.

considered IC (in-class) if at least one annotator labeled the image a clear sample of the corresponding class

For the pre-filtering strategy (ii) and for the selection of easy samples (iii), we compute text-alignment using CLIP score and we remove all samples that have a CLIP similarity $s_{\text{CLIP-text-alignment}} > 24$, which approximately includes 90% of all ImageNet validation images [60]. We use the implementation in *torchmetrics* with VIT-B/16. After removing the easy samples in step (iii), 2.7k images remain for labeling. We use the VIA annotation tool [12, 13] to create the annotations. Each image is labeled by two humans. In total, 14 graduate students are involved in the labeling process. For all participants, we ensure sufficient motivation and they receive detailed instructions on how to perform the labeling (the full set of instructions is provided in Fig. 33). We provide the filtering statistics in Tab. 8 and the statistics of the labeled dataset in Fig. 28. An example screenshot of the labeling tool is visualized in Fig. 27.

J. User Study

We perform a user study on the final dataset using the same tooling as for the human labeling discussed in Appendix I (iv). The user study includes 300 randomly sampled images from the benchmark and it is checked by two different individuals. In total, the user study involved seven people with different professions. 3 samples of our benchmark were considered as out-of-class samples, resulting in a ratio of 1% of failure cases with a margin of error of 0.5% for a one-sigma confidence level.

Table 9. **User study shift realism.** Distribution of images where the shift is clearly identifiable.

Scale	Unclear	Clear
1.0	0.76	0.24
1.5	0.51	0.49
2.0	0.24	0.76
2.5	0.16	0.84

Figure 29. **Combination of Sliders.** We exemplarily show that sliders can be combined. Here, a snow slider (vertical axis) and a cartoon slider (horizontal axis) are linearly added for three scales.

We also study when a shift is clearly visible and report it in Tab. 9. Model performance is evaluated only for 030 seeds where all scales are valid.

K. Applications of Trained Sliders

We can combine various sliders by simply adding the corresponding LoRA adapters. We show an example application in Fig. 29.

L. OOD-CV Details

The Out-of-Distribution Benchmark for Robustness (OOD-CV) dataset includes real-world OOD examples of 10 object categories varying in terms of 5 nuisance factors: *pose*, *shape*, *context*, *texture*, and *weather*.

Generation of images for synthetic OOD-CV We generate the images for the synthetic OOD-CV dataset using a larger number of noise steps (85%) and more scales (between 0 and 3). The shift sliders for these classes appear to be more robust potentially since these classes occur more often in the dataset for training CLIP and Stable Diffusion. We use SD2.0 to generate the images.

Table 10. **OOD-CV Statistics.** We report the number of images and accuracies for the weather subset.

Shift	#images	Accuracy
Snow	273	70.3
Fog	24	62.5
Rain	74	66.2
Unknown	129	66.7
Total	500	68.4

Training subset The OOD-CV benchmark provides a training subset of 8627 images. We train various classifiers (i.e., ResNet-50 [21], ViT-B/16 [10], and DINO-v2-ViT [45]) for classification. We finetune each baseline during 50 epochs with an early stopping set to 5 epochs. We apply standard data augmentations such as scale, rotation, and flipping during training. The training subset is composed of images originating from different datasets, notably ImageNet [8] and Pascal-VOC [15]. It is important to notice that the distribution of these two subsets is slightly different, with a higher data quality for the ImageNet subset and a lower quality for the latter subset (more noise, smaller objects, different image sizes). We visualize a few examples of the training data in Fig. 32.

Test subset annotations In the test subset provided in the benchmark dataset, only the coarse individual nuisance factors (*e.g.*, *weather*, *texture*) are provided. In our setup, we are interested in studying more fine-grained nuisance shifts, notably *rain*, *snow*, or *fog*. Hence, we had to assign some fine-grained annotation to all images containing *weather* nuisance shifts. Hence, we assign a fine-grained annotation by computing the CLIP similarity to the following texts: "a picture of a `class` in `shift`", where `class` is the ground truth class and `shift` the nuisance shift candidate *rain*, *snow*, or *fog* and "a picture of a `class` without snow nor fog nor rain". By applying a softmax on the similarity scores with the previous texts, we can assign the fine-grained nuisance shift *rain*, *snow*, *fog* or *unknown* for each image. We show more statistics in Tab. 10. By checking the results visually, we observe that all fine-grained nuisance shifts align with human perception and have a tendency towards classifying samples as *unknown* as soon as there is a small doubt. Note that by applying the same strategies to our generated data, we obtain an accuracy close to 100%.

Nearest neighbor images of OOD-CV and CNS-Bench. To illustrate the realism of our generated image, we compute the nearest neighbours using cosine similarity with CLIP image embedding and we plot it in Fig. 31.

Failure point distribution for CNS-Bench (OOD-CV) Fig. 30 depicts the failure distribution for the three shifts.

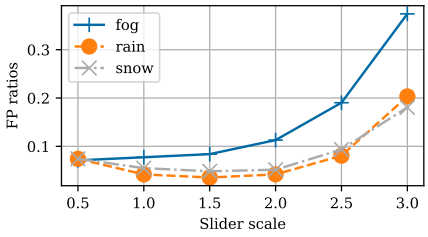


Figure 30. **Failure point distribution of a ResNet-50 classifier on our continuous OOD-CV benchmark.** Our benchmark allows computing the failure distribution of failure points, allowing the analysis of when classifiers tend to fail, which was not possible using the manually labeled images.

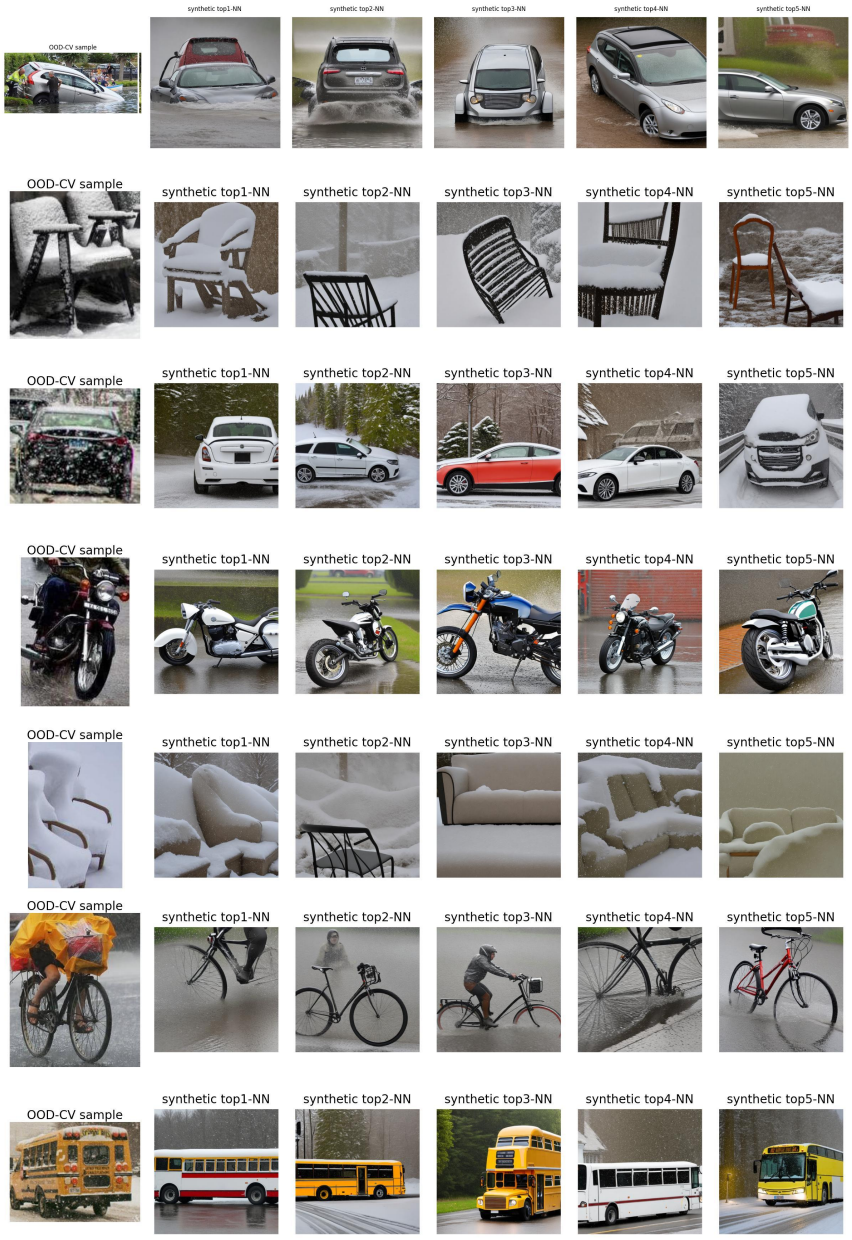


Figure 31. **Closest synthetic samples to two example OOD-CV images.** We find the top-5 nearest neighbours using cosine similarity with CLIP image embedding.

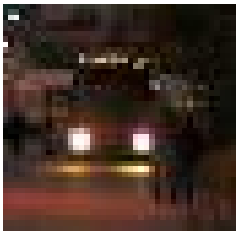
(a) Train, ImageNet.

(b) Train, ImageNet.

(c) Train, ImageNet.

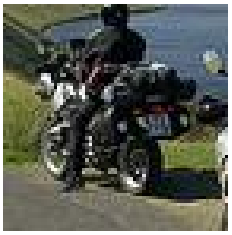
(d) Train, ImageNet.

(e) Train, Pascal-VOC.



(f) Train, Pascal-VOC.

(g) Train, Pascal-VOC.



(h) Train, Pascal-VOC.

(i) Test, snow shift.

(j) Test, snow shift.

(k) Test, snow shift.

(l) Test, rain shift.

Figure 32. **OOD-CV example images.** We illustrate a set of example images from the training and the testing dataset of OOD-CV: (a-h) example from the training set, from ImageNet or Pascal-VOC. (i-l) Some examples for weather nuisance shifts. In the training set, we observe that images from the Pascal-VOC subset are usually of lower quality (*e.g.*, cropping, occlusion, resolution) compared to the ImageNet subset. In the test set, we see that they are not fully disentangled (*e.g.*, (j) is only partially visible, (k) is partially occluded).

Labeling task for out-of-class detection

Motivation: For benchmarking a classifier with synthetic images, we need to ensure that the generated images still correspond to the correct classes. To evaluate automatic filtering pipelines, we create a dataset with human labels. The dataset includes generated images with various levels of snow or cartoon style.

Task:
The goal is to detect images that do not belong to the corresponding ImageNet class (given as title).

Given an image, your task is to select one of three labels:

- ***class***:
  - You can clearly recognize the class.
- ***partly class***:
  - Given the class label, the class seems to correspond to the image.
  - You can recognize parts of the class but you are not very sure whether this is actually the class
  - You clearly see some characteristics of the class but it does not include all the important features.
- ***not class***:
  - The considered image is clearly not the considered class.

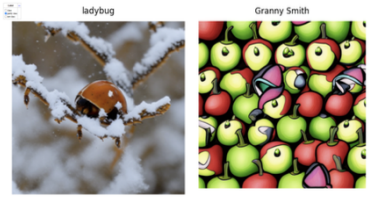
The goal is to check whether the objects in the image correspond to a class or not. The goal is not to check whether the samples look realistic.

Every class starts with one realistic example image, taken from ImageNet. This image needs to be labeled as well. Since the example is just one illustrative example, not depicting the diversity of the class, it is recommended to use Google picture search to get an intuition of how the object looks in case one is not familiar with the class.
Some of the consecutive class samples will be similar. They are generated with the same seed but with varying snow or cartoon levels.

Some examples for class, partly class, and not class:

1) ***class***: This animal can be clearly described as a fox at first glance. Also, the bucket can be easily recognized.

2) ***partly class***: The shape and size seems to fit a ladybug. However, the black dots are missing. The other picture might be a cartoon-like illustration of apples. However, this can be argued. It is not clear.



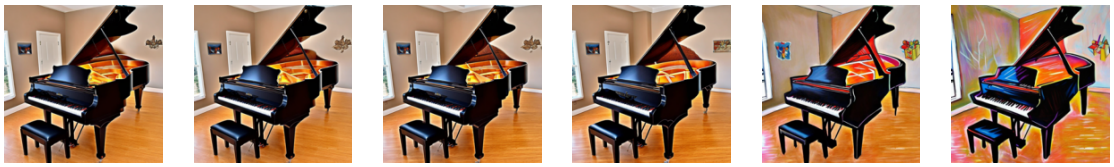
3) ***not class***: First example: This is supposed to be a sax but it is clearly not recognizable as a sax. Second example: There is not a single characteristic that resembles a hammerhead. It is very clearly not the class.

Figure 33. **Set of instructions for labeling.** Instructions provided to the human annotators to perform the labeling of the out-of-class filtering dataset.

(a) Style of a tattoo.

(b) Cartoon style.

(c) Style of a video game.



(d) Graffiti style.

(e) Painting style.

(f) Pencil sketch style.

(g) Plush toy style.

(h) Design of a sculpture.

Figure 34. **Example sliding for various nuisance shifts.** We visualize six generated images with the corresponding scales as 0, 0.5, 1, 1.5, 2, and 2.5.

(a) In heavy snow.

(b) In a sandstorm.

(c) In dust.

(d) In smog.

(e) In fog.

(f) In heavy rain.

Figure 35. **Example sliding for various nuisance shifts.** We visualize six generated images with the corresponding scales as 0, 0.5, 1, 1.5, 2, and 2.5.

M. Datasheet

In the following, we answer the questions as proposed in Gebru et al. [19].

M.1. Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

The dataset was created to evaluate the robustness of state-of-the-art models to specific continuous nuisance shifts. Current approaches are not scalable and often include only a small variety of nuisance shifts, which are not always relevant in the real world. More importantly, current benchmark datasets define binary nuisance shifts by considering the existence or absence of that shift, which may contradict their continuous realization in real-world scenarios.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The paper was created by the authors of the CNS-Bench paper, which are affiliated with the listed organizations.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

The creation was funded by by the German Science Foundation (DFG) under Grant No. 468670075.

M.2. Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?

The dataset consists of synthetic images that were generated using Stable Diffusion.

How many instances are there in total (of each type, if appropriate)?

The dataset contains $192,168$ images in total, with $32,028$ for each of the six scales with 14 shifts. Each shift has at least $5,000$ images and 100 classes.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances because instances were withheld or unavailable).

The dataset contains the subset of images that were filtered using the selected filtering strategy. Originally, $420,000$ images were generated.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

"Raw" synthetically generated data as described in the paper.

Is there a label or target associated with each instance? If so, please provide a description.

Yes, each image belongs to an ImageNet class and has a shift scale assigned to it.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

No, for each instance, we give the class label, the shift and its scale, and the parameters used for generating this image. However, the class label might be erroneous in rare cases where the generated image corresponds to an out-of-class sample.

Are relationships between individual instances made explicit (e.g., users with their tweets, songs with their lyrics, nodes with edges)? If so, please describe how these relationships are made explicit.

Yes, the relationships in terms of class, random seed for generation, shift, and scale of shift are provided in the dataset.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

We offer a benchmark dataset specifically intended for testing the robustness of classifiers. Therefore, we recommend utilizing the entire dataset provided as the test dataset.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

We provided a dataset of generated images. While we apply a filtering strategy to reduce the number of out-of-class and unrealistic samples, we cannot guarantee that all images of the dataset represent a realistic and visually appealing realization of the considered class. We provide a statistical estimate of the number of failure samples in the paper. The data might also include the redundancies that underlie the image generation process of Stable Diffusion.

Is the dataset self-contained, or does it link to or

otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with the use of these external resources?

The dataset is fully self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.

No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

There is a small chance that our synthetically generated data can generate offensive images. However, we did not encounter any such sample during our extensive manual annotations.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

No.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

N/A.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

N/A.

Does the dataset contain data on individuals' protected characteristics (e.g., age, gender, race, religion, sexual orientation)? If so, please describe this data and how it was obtained.

N/A.

Does the dataset contain data on individuals' criminal history or other behaviors that would typically be considered sensitive or confidential? If so, please describe this data and how it was obtained.

N/A.

M.3. Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses)?

N/A.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

We used Stable Diffusion 2.0 to generate all images. Images were generated using NVIDIA A100 and A40 GPUs.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

The dataset was filtered using a combinatorial selection approach using the alignment scores of DINOv2 and CLIP to the considered class.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

The authors of the paper and other PhD students of the institute. They were not additionally paid for the dataset collection process.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

The images were generated and processed over a timeframe of four weeks.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No ethical concerns.

M.4. Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so,

please provide a description. If not, you may skip the remaining questions in this section.

Yes, cleaning of the generated data was conducted. The generated images underwent filtering to reduce the number of out-of-class samples using the proposed filtering mechanisms. Instances that did not meet these criteria were removed from the dataset. For a detailed description of the filtering process, please refer to the corresponding section in the paper.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

The generated images remain in their original, unprocessed state and can be considered as "raw" data. However, we have not provided all the images that were filtered out.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

Generating the images was performed using commonly available Python libraries. For annotating a subset of the dataset for filtering purposes, we have used the VIA annotation tool [12, 13].

M.5. Uses

Has the dataset been used for any tasks already? If so, please provide a description.

In our work, we demonstrate how this approach yields valuable insights into the robustness of state-of-the-art models, particularly in the context of classification tasks.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

The relevant links can be acquired via the project page https://genintel.github.io/CNS.

What (other) tasks could the dataset be used for?

Our work showcases the capability of our dataset to enhance control over data generation, which is particularly evident through continuous shifts. However, its applicability extends beyond this demonstration. The dataset can be effectively utilized in various generation tasks that necessitate continuous parameter control. While we showcased its efficacy in providing insights for models tackling classification tasks, it can seamlessly extend to evaluate the robustness of state-of-the-art methods across diverse tasks such as segmentation, domain adaptation, and many others. This is possible by combining our approach with other modes of conditioning Stable Diffusion. In addition, our data can also be used for fine-tuning, which we also demonstrated in the supplementary material.

Is there anything about the composition of the dataset or the way it was collected and cleaned that might impact future uses? For example, is there anything that might cause the dataset to be used inappropriately or misinterpreted (e.g., accidentally incorporating biases, reinforcing stereotypes)?

Our dataset was synthesized using a generative model. It, therefore, likely inherits any biases for its generator. Similarly, filtering is performed by pre-trained models, which can indirectly also contribute to biases.

Are there tasks for which the dataset should not be used? If so, please provide a description.

No, there are no tasks for which the dataset should not be used. Our dataset aims to enhance model robustness and provide deeper insights during model evaluation. Therefore, we see no reason to restrict its usage.

M.6. Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

Yes, the dataset will be publicly available on the internet.

How will the dataset be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?

The dataset will be distributed as archive files on our servers.

When will the dataset be distributed?

The dataset will be distributed upon acceptance of the manuscript.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU.

CC-BY-4.0.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access

point to, or otherwise reproduce, any relevant licensing terms.

No, there are no IP-based or other restrictions on the data associated with the instances imposed by third parties.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

We are not aware of any export controls or other regulatory restrictions that apply to the dataset or to individual instances.

M.7. Maintenance

Who is supporting/hosting/maintaining the dataset?

The dataset is supported by the authors and their associated research groups. The dataset is hosted on our own servers.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

The authors of this dataset will be reachable at their e-mail addresses.

Is there an erratum? If so, please provide a link or other access point.

If errors are found, an erratum will be added to the website.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, when, and how updates will be provided.

Yes, updates will be communicated via the website. The dataset will be versioned.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a specific period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

Our dataset does not relate to people.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how.

No, older versions of the dataset will not be supported if the dataset is updated. We do not plan to extend or update the dataset. Any updates will be made solely to correct any hypothetical errors that may be discovered.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be made publicly available?

Yes, we provide all the necessary tools and explanations to enable users to build continuous shifts for their own specific applications. Our dataset serves as a foundation to evaluate various classifiers. We encourage to build on top of this work and we are happy to link relevant works via our GitHub page.

M.8. Author Statement of Responsibility

The authors confirm all responsibility in case of violation of rights and confirm the license associated with the dataset and its images.