Appendix:
COCOA: Context-Conditional Adaptation for Recognizing Unseen Classes in Unseen Domains

In this appendix, we discuss the following details, which could not be included in the main paper owing to space constraints:

- Propose a new fine-grained ZSLDG benchmark - *CUB-Corruptions* and evaluate performance of COCOA on it along with component-wise analysis (in continuation to Sec 4)
- Implementation details for each stage of our proposed methodology to facilitate reproducibility
- Additional analysis of performance on DomainNet dataset: (1) Performance analysis with varying number of synthesized unseen class features; (2) t-SNE plots for more classes following Sec 4 of main manuscript
- Results on standard ZSL and DG setting usings our proposed approach (in continuation to results in Sec 4)

A. CUB-Corruptions: Dataset and Results

Given that DomainNet dataset is the only and primary benchmark dataset for the ZSLDG as studied in [A4], we propose a new benchmark - *CUB-Corruptions* - which is a relatively small-scale but fine-grained dataset unlike DomainNet (which is large-scale and coarse grained). This benchmark facilitates holistic evaluation of models on dataset with varying granuality and faster prototyping (due to a relatively smaller size). This benchmark is inspired by the general common-corruptions benchmark for image classification [A1] which closely resembles real-world conditions and incorporates domain shifts that are often encountered practically (eg. *snow*, *fog*, *blur* etc.). Besides the domain shift, the corruptions also obscure objects and class-level information (unlike DomainNet where domain difference refers mostly different ways to depict the same object such as *clipart, sketch, painting etc*), thus making it challenging for the model to generalize at test-time. We now describe the dataset and subsequently, our results on the benchmark.

(Proposed) CUB-Corruptions Benchmark. This dataset is obtained by applying corruptions like *motion blur*, *spatter*, *snow*, *fog*, *pixelate* which are frequently found in real-world scenarios [A1] on the well-known CUB dataset [A7] to establish the first fine-grained dataset for the ZSLDG problem setting. This provides us with images in 6 different domains (including the *real* image domain). For the ZSL part, we follow the standard zero-shot splits on CUB for seen and unseen classes as proposed in [A8]. Also, we maintain a constant severity level of 5 [A1] for each of these corruptions. Figure A shows sample images from the dataset. Similar to the training protocol followed for

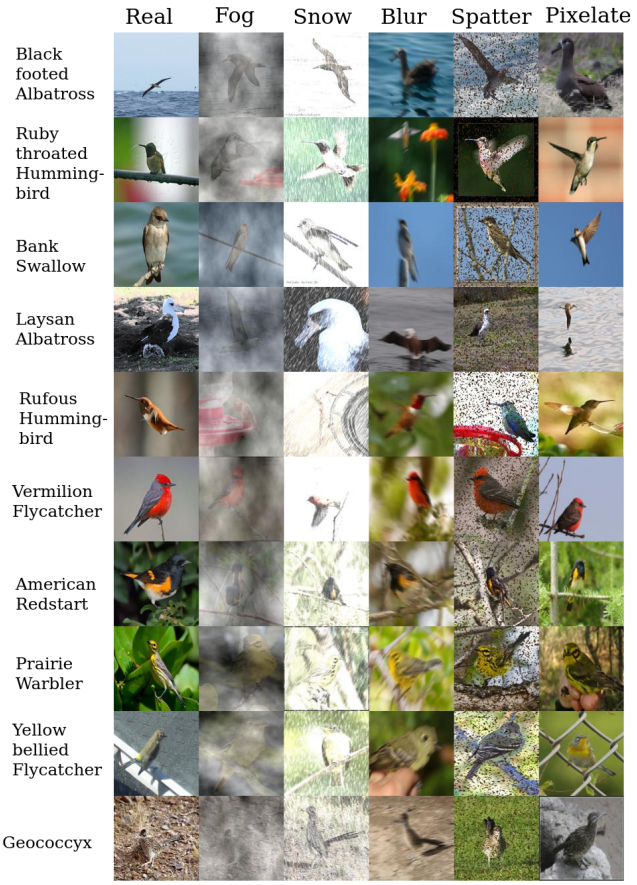

Figure A: Samples from *CUB-Corruptions* dataset. Rows correspond to classes and columns correspond to domain-shifts.

Method	Target Domain					
	Fog	*Snow*	*Blur*	*Spatter*	*Pixelate*	Avg.
Mixup Img-only	27.6	21.5	39.38	24.5	49.5	32.496
Mixup two-level	32.2	26.4	41.8	28.38	56.5	37.056
CuMix	31.8	25.33	42.24	30.3	58	37.534
f-clsWGAN	25	21.66	28.77	28.47	40.22	28.824
AGG + f-clsWGAN	32.12	31.22	43.4	32.43	57	39.234
CuMix + f-clsWGAN	33.63	30.51	42.4	32.92	56.11	39.114
ROT + f-clsWGAN	34.57	31.11	43.61	33.9	58.53	40.344
$COCOA_{DOM(0.1)}$	34.1	30.8	42.9	30.88	59.34	39.604
$COCOA_{AGG}$	34.33	31.7	44.1	31.33	**60.26**	40.344
$COCOA_{CuMix}$	**36.36**	<u>31.66</u>	<u>44.58</u>	<u>35.31</u>	59.45	<u>41.472</u>
$COCOA_{ROT}$	<u>35.56</u>	**34.48**	**46**	**37.6**	<u>60.07</u>	**42.742**

Table A: Performance comparison with state-of-art methods for ZSLDG problem setting on the proposed CUB-Corruptions benchmark. All reported results follow the protocol and splits as described in Sec A. Best results highlighted in bold and second best are underlined.

DomainNet, we train on 5 domains and test on the left-out domain, and also present the average results obtained by changing the target domain in each run. For semantic representations, we use the 312-dimensional semantic attributes provided with the CUB dataset, commonly used in standard ZSL work [A8].

Evaluation on CUB-Corruptions Benchmark. Table A shows the results on the CUB-Corruptions benchmark. We notice a similar trend in the average performance across domains as observed in the case of DomainNet. Our method variant $COCOA_{ROT}$ establishes a new state-of-the-art for the proposed dataset. Our method outperforms CuMix [A4] by a margin of 5 % average across domains (i.e 25 % relative improvement) implying that our model is more robust to corruption based domain-shifts commonly encountered in practical real world scenarios. The proposed method is also superior to standard ZSL methods like f-clsWGAN and its combinations with different visual backbones. (Refer to baseline section in main manuscript for more details).

We also show component-wise analysis of our method on the CUB-Corruptions dataset in Table B. Note that the variants (*i.e S1, S2, S3, S4*) follow the same definition as established for DomainNet component analysis in Sec 4 of main manuscript.) Once again, as in Sec 4, $S4$ (our final overall framework) provides the best results.

Variant	*Fog*	*Snow*	*Blur*	*Spatter*	*Pixelate*	Avg.
$S1$	28.75	27.7	40.86	28	53.4	35.742
$S2$	36.5	33	44.8	34.8	60.62	41.944
$S3$	35.36	34.9	45	35.5	60	42.152
$S4$	35.56	34.48	46	37.6	60.07	42.742

Table B: Ablation study for different components of our framework on CUB-Corruptions dataset

B. Implementation Details

In this section, we describe the module-wise implementation details of our method to facilitate reproducibility.

Feature Extraction Module. The first step in our approach is to train a generalizable feature extractor $f(.)$. Following previous works [A4], we choose a Resnet-50 architecture as our visual encoder. For rotation-based self-supervised regularization used in $COCOA_{ROT}$, we perform a 4-way classification as auxiliary task in $\mathcal{L}_{ROT}$ where the set of allowed rotations considered are $\mathcal{S}^{ROT} = \{0°, 90°, 180°, 270°\}$. Both semantic projector $p(.)$ and auxiliary rotation classifier are implemented as single linear layer networks. Again following previous works [A4], we use SGD with momentum optimizer, with a learning rate of 0.001 for semantic projector $p(.)$ as well as auxiliary rotation classifier with a weight decay of $5 \cdot 10^{-5}$ and momentum 0.9. Training is done for 8 epochs, with a batch-size containing 24 samples per domain (total batch-size 120). Learning rates are decreased by a factor of 10 after 6 epochs.

Generative Module. The second step of our approach involves training the generator which is integrated with context-conditional BatchNorm to generate features matching the distribution of real features $\mathbf{f}$, extracted in the previous stage. We implement both generator $G(.)$ and discriminator $D(.)$ as 4-layer MLPs with hidden layer size as 4096 and 2048 in case of DomainNet and CUB-Corruptions respectively. Leaky ReLU activation is used for intermediate layer representations and sigmoid for final layer representations. Domain-specific embeddings $\mathcal{E}^{gen}$ and $\mathcal{E}^{disc}$ have dimensions of size 64. We use Adam optimizer with learning rate 0.0001 for both discriminator and generator networks, batch-size of 500 and noise dimension of 312 (for both CUB-Corruptions and DomainNet). The network is

trained for 70 epochs, and in each mini-batch iteration, the discriminator is updated 5 times before updating generator.

Recognition/Inference Module. After training the generator and discriminator in the previous stage, we generate N features for each unseen class using semantic representation and interpolated domain embeddings, $\mathcal{E}^{gen}_{interp}$ (generated using source embeddings $\mathcal{E}^{gen}$ obtained in the previous stage). In case of DomainNet, N is chosen to be $2000, 1000, 1500, 1000, 1000$ when held out test-domains are *painting, quickdraw, infograph, sketch and clipart* respectively. In case of CUB-Corruptions, N is set to 1500 for all held-out domains. The classifier $\mathcal{C}(.)$ is implemented as a single linear layer following standard protocol and is trained for 25 epochs using Adam optimizer with learning rate 0.001, $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

C. Additional Analysis on DomainNet

Varying number of synthesized features. Herein, we study the performance of COCOA with varying number of synthesized features N for unseen classes. To analyze the performance for different levels of domain shift at test-time, we plot these results for a relatively easier (*i.e painting*) and harder domain (*i.e quickdraw*) in Figure B. We observe that the best results are obtained in the range of $N = 1000 - 2000$ features and the performance drops as we further increase the generated features. It can be clearly seen that the trend for ZSLDG accuracy is fairly stable with variation in number of synthesized features for DomainNet dataset, supporting the robustness of our algorithm to such hyperparameter choices. Note that we found that other domains of the DomainNet dataset also followed a similar trend.

t-SNE plot visualization. Figure C shows t-SNE plot visualizations of generated features for more classes from DomainNet (source domains - *Real*, *Infograph*, *Quickdraw*, *Clipart*, *Sketch*) when only semantic representations are used in context vector (top row) and when both semantic and domain embeddings are concatenated to get context vector (bottom row). Note that the classes are selected randomly as mentioned previously. We notice the same observation as earlier that when both semantic and domain context are used (Fig C, Row 2), the model better captures the necessary data distribution.

D. Performance on Standard ZSL and DG Settings

For completeness, we also study the effectiveness of our proposed approach on the standard ZSL and DG settings and compare our performance to the state-of-the-art ZSLDG method CuMix [A4].

Switching Context. For the standard DG setting, the "context" vector would correspond only to domain-specific information (since classes at both test time and training belong to the same categories in this setting). Thus, we only use domain embeddings in the context vector, $\mathbf{c}$, as input to the BatchNorm estimator. We synthesize visual features across domains (as in Sec 3.4 of main manuscript) to train the final classifier. Since DG setting doesn't assume access to semantic representations, we use one-hot vectors (as semantic representations).

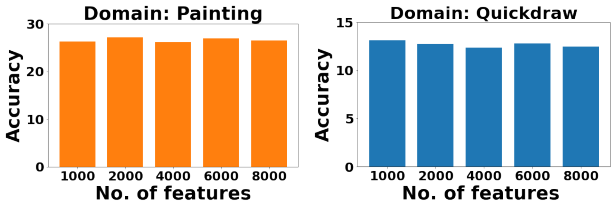

Figure B: Variation in ZSLDG performance with different number of synthesized features for unseen classes

On the other hand, for the standard ZSL setting, "context" refers to only class-level semantics, since images during train and test-time belong to the same domain. Thus, the context vector input to the BatchNorm estimator comprises of the semantic representation only i.e $\mathbf{c} = \mathbf{a}^u_y$.

Results. Following [A4], we evaluate the performance for standard DG setting on the well-known PACS dataset [A3] using ResNet-18 backbone in Table C. In Table D, we present the results on the standard ZSL setting on SUN (large-scale, coarse) [A6], FLO (fine-grained) [A5] and AWA2 (coarse-grained) [A2] datasets. For the ZSL experiments, we use ResNet-101 features – extracted from a pre-trained ImageNet model – to train the generative model, following [A4]. Note that we follow standard protocols and backbones for all our experiments to facilitate fair comparison. From both tables, it can be clearly seen that even though our method is designed to address the ZSLDG problem setting, it demonstrates competitive performance on standard ZSL and DG settings (improved performance when compared to previous ZSLDG state-of-art approach, CuMix). This demonstrates the usefulness of the context vector in our approach. By switching the context appropriately, our method tackles standard DG, standard ZSL and the harder ZSLDG settings using the same framework.

References

[A1] Dan Hendrycks and Thomas G. Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *ArXiv* abs/1903.12261 (2019).

[A2] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. "Learning to detect unseen object classes by between-class attribute transfer". In: *2009*

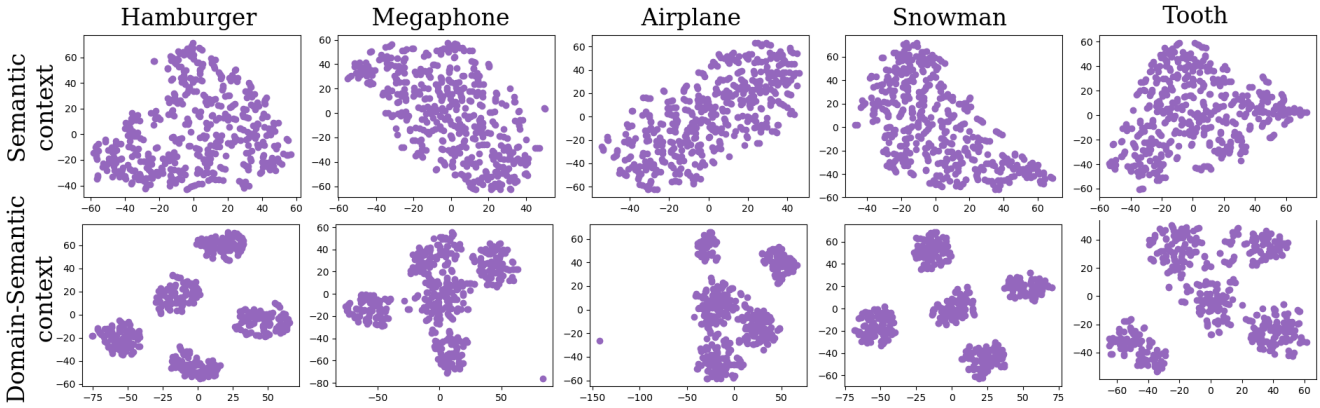

Figure C: t-SNE visualizations of image features synthesized by COCOA for randomly selected unseen classes (*Hamburger*, *Megaphone*, *Airplane*, *Snowman*, *Tooth*) using only semantic context *(Row 1)* and both domain-semantic context *(Row 2)* *(Best viewed in color, zoomed in)*

Model	*Photo*	*Art*	*Sketch*	*Cartoon*	Avg
AGG	94.9	76.1	69.4	73.8	78.5
DANN (JMLR'16)	94.0	81.3	73.8	74.3	80.8
MLDG (AAAI'18)	94.3	79.5	71.5	77.3	80.7
CrossGrad (ICLR'18)	94.0	78.7	65.1	73.3	80.7
MetaReg (NeurIPS'18)	94.3	79.5	72.2	75.4	77.8
D-SAM (GCPR'18)	95.30	77.33	77.83	72.43	80.72
JiGen (CVPR'19)	96.0	79.4	71.4	75.3	80.4
Epi-FCR (ICCV'19)	93.9	82.1	73.0	77.0	81.5
MASF(NeurIPS'19)	94.99	80.29	71.69	77.17	81.04
MMLD (AAAI'20)	**96.1**	81.28	72.29	77.16	81.83
DMG (ECCV'20)	93.35	76.90	75.21	80.38	81.46
CuMix (ECCV'20)	95.1	82.3	72.6	76.5	81.6
BNE (CVPR'21)	94.8	78.8	**79.7**	78.9	83.1
COCOA$_{ROT}$(*Ours*)	95.61	**83.15**	76.2	**80.46**	**83.86**

Table C: Standard DG performance of COCOA on PACS using ResNet-18 backbone. We follow the standard training/validation/testing splits and evaluation protocols as in [A3]. Best results are highlighted in bold and second best results are underlined.

Method	AWA2	FLO	SUN
DEVISE(NIPS 2013)	59.7	45.9	56.5
CONSE(ICLR 2014)	44.5	-	38.8
SSE(ICCV 2015)	61.0	-	51.5
LATEM(CVPR 2016)	55.8	40.4	55.3
SYNC(CVPR 2016)	54.0	-	56.3
ALE(TPAMI 2016)	62.5	48.5	58.1
SJE(CVPR 2015)	61.9	53.4	53.7
ESZSL(ICML 2015)	58.6	51.0	54.5
SYNC(CVPR 2016)	46.6	-	56.3
SAE(CVPR 2017)	54.1	-	40.3
CVAE-ZSL(CVPRW 2018)	65.8	-	61.7
SP-AEN(CVPR 2018)	58.5	-	59.2
f-CLSWGAN(CVPR 2018)	**68.2**	67.2	60.8
CADA-VAE(CVPR 2019)	64	-	61.8
CuMix(ECCV 2020)	64.0	59.7	62.4
COCOA(*ours*)	65.9	**67.6**	**62.78**

Table D: Standard ZSL performance of COCOA on AWA2, FLO and SUN. we follow the standard training/validation/testing splits and evaluation protocols as in [A8]. Best results are highlighted in bold and second best results are underlined.

IEEE Conference on Computer Vision and Pattern Recognition. 2009, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594.

[A3] Da Li et al. "Deeper, Broader and Artier Domain Generalization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5543–5551. DOI: 10.1109/ICCV.2017.591.

[A4] Massimiliano Mancini et al. "Towards Recognizing Unseen Categories in Unseen Domains". In: *ECCV*. 2020.

[A5] 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*. 2008, pp. 722–729. DOI: 10.1109/ICVGIP.2008.47.

[A6] Genevieve Patterson and James Hays. "SUN attribute database: Discovering, annotating, and recognizing scene attributes". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*.

2012, pp. 2751–2758. DOI: 10.1109/CVPR.2012.6247998.

[A7] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[A8] Yongqin Xian, Bernt Schiele, and Zeynep Akata. "Zero-Shot Learning - The Good, the Bad and the Ugly". In: *IEEE Computer Vision and Pattern Recognition (CVPR)*. 2017.