Appendix

A. Full ACE results

As described in Section 5.1, we evaluated DNNs on datasets generated using each GCM (see Figure A1). For each set of results, we run CDRA to estimate the per variable $ACE_M(V : v \rightarrow \tilde{v})$. The full set of results for all GCMs and imaging factors can be seen in Table A2. Average error (Δ_{ACE}) is found in Table 3. We also show the per-GCM mean Top-1 accuracy in Table A4 as well as the single node Top-1 accuracy for level 1 corruption severity in Table A5.

These results provide several insights (consistent with those discussed in the main manuscript). First, we see that mean accuracy alone gives little information about what factors affecting image quality have the strongest effect on DNN accuracy. The mean accuracies per GCM indicate that under compounded corruptions, task DNN accuracy is significantly degraded. However, from mean accuracy alone, we cannot interpret which factors of the imaging process may be driving this substantial decrease. In contrast, CDRA is able to accurately identify the sensitivities of DNNs to specific imaging factors grounded by knowledge or informed assumptions about the imaging domain (in the form of the causal DAG).

Second, we see that the value of the individual ACE values is strongly linked to the topology of the DAG. For example, the presence of impulse noise (IN) occurs in a majority of the GCMs, and across these models, ACE values range from ≈ -13 up to ≈ -3. We also observe that ACE_{acc} for the single factor GCMs (which represent the current state-of-practice) is not predictive of ACE_{acc} for the more complex GCMs (*e.g.* compare IN in Table A3 with IN in Table A1). Since the error analysis shows that our estimates $\widehat{ACE}_M(V : v \rightarrow \tilde{v})$ are close to the ground truth values, we can see that the single factor GCMs are in general not informative of DNN sensitivities in more complex real-world domains. These results further underscore that when knowledge of the specific imaging domain is available, it can be used to effectively estimate fine-grained DNN sensitivities.

Lastly, we re-emphasize that we can run CDRA directly on complex image data and still obtain accurate estimates of the $ACE_M(V : v \rightarrow \tilde{v})$ values. Each GCM in this experiment produces a wide diversity of imaging conditions and compounded corruptions that prior work is able to evaluate effectively.

B. Full DAG misspecification results

Table B7 contains the full set of results illustrating how ACE estimation error changes as a function of misspecifications of the GCM DAG. Both tables show how much the estimation error deviates from the baseline estimation error and each cell is an average over all factors in the GCM. The results show that misspecifications of the DAG contribute less than 1% additional error to the ACE estimates. The mean and standard deviation of this residual error increase slightly as the degree of DAG misspecification error increases, but the total ACE error in the most extreme cases is still relatively small.

C. Causal identification example

We provide here an example of the identification process for variables in GCM 0 shown in full in Figure C2. Table C8 provides the set of variables identified as the adjustment set $\mathcal{W}$ using the backdoor criterion to be included as input to the $\widehat{ACE}_M(V : v \rightarrow \tilde{v})$ estimator (*e.g.* $\hat{\mu}(w, v)$ from Sec. 3.2).

D. Sample images from GCMs

Figure D3 shows examples of images sampled from the GCMs used for the experiments in Sections 5.1 and 5.2. The examples in Figure D3 are sampled randomly from the full set of 50k images and rendered using the process described in Section 4. Note that the imaging conditions here are more complex than if only a single factor is applied (*i.e.* the approach used by the common corruptions framework), yet the images are still interpretable. The low mean accuracies in Table A4 show that these compounded corruptions have a significant impact on DNN performance despite the adequate interpretability of the images.

E. Rendered corruptions

The following describes how values from the GCM in Section 5.3 were sampled, normalized, and used to render CLEVR and MOVi-C variants. Because the imaging factors correspond to continuous values in Blender, we specify functional relationships between various factors in the GCM. For each factor A, the normalized factor value is first computed as

$$Z = \sum_{A' \in pa(A)} \alpha_{(A',A)} \cdot V_{A'} + U_A \tag{1}$$

$$V_A = \beta_A^{-1}(f(Z)) \tag{2}$$

where $\alpha_{(A',A)}$ is the associated directed edge weight for the edge (A', A), $V_{A'}$ the sampled value for the parent factor A', $f : \mathbb{R} \rightarrow [0, 1]$ a normalization function, and $\beta_A^{-1}(Z)$ the inverse CDF of a random variable $Z \sim \beta(a_A, b_A)$, and $U_A \sim N(0, \sigma_A)$ is an exogenous noise term. For experiments in Sec. 4.4, the edge weights were each randomly sampled from $\mathcal{U}(-1, 1)$.

Normalized factor values are mapped back to Blender settings to enable physics-based scene rendering. The final Blender setting is calculated for each factor. For *increasing* factors, the normalized V_A is rescaled to $[\min_A, \max_A]$

Table A1. **True** $ACE_{acc}(V\!:\!0\!\rightarrow\!1)$ (%) **calculated for each GCM, imaging factor, and task DNN.** Each value represents the expected change in accuracy as a result of increasing the corruption severity associated with the corresponding variable. Note how ACE changes for different nodes as a function of the DAG topology. Values close to 0 (or > 0) indicate higher robustness.

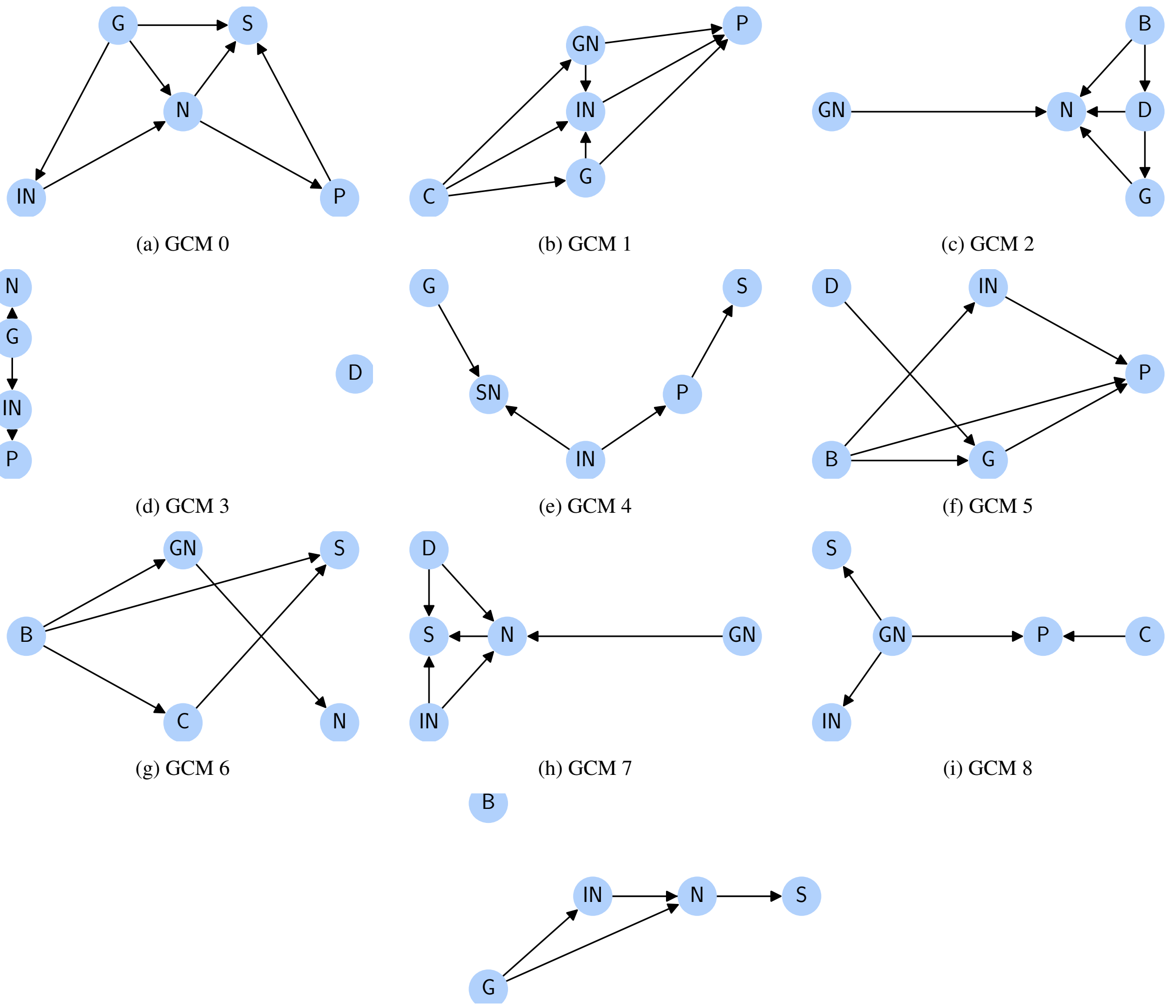
GCM	0					1					2				
DNN / Factor	G	IN	N	P	S	C	G	GN	IN	P	B	D	G	GN	N
ConvNext-B	-5.3	-4.5	-3.4	-6.9	-8.1	-22.5	-2.4	-4.9	-12.5	-2.4	-0.64	-8.3	-3.3	-1.8	-6.3
ResNet50	-7.0	-7.0	-3.6	-5.3	-11.5	-27.2	-4.8	-5.9	-12.4	-3.9	0.04	-12.4	-4.7	-4.1	-13.6
Swin-B	-4.8	-4.9	-4.0	-9.1	-7.3	-17.1	-3.8	-4.7	-9.9	-8.3	0.44	-9.1	-3.2	-1.8	-4.6
GCM	3					4					5				
DNN / Factor	D	G	IN	N	P	G	IN	P	S	SN	B	D	G	IN	P
ConvNext-B	-5.3	-3.4	-5.0	-2.9	-4.2	-4.1	-4.6	-12.6	-9.6	-5.5	-1.4	-5.4	-0.35	-3.5	-6.4
ResNet50	-8.0	-5.9	-8.6	-4.1	-0.26	-4.7	-7.6	-13.6	-11.0	-8.6	-1.7	-7.3	0.22	-6.3	-8.2
Swin-B	-5.7	-2.9	-5.1	-3.7	-6.2	-3.3	-4.7	-15.2	-8.4	-4.8	-1.1	-4.7	-0.46	-4.2	-6.9
GCM	6					7					8				
DNN / Factor	B	C	GN	N	S	D	GN	IN	N	S	C	GN	IN	P	S
ConvNext-B	-8.5	-28.9	-10.8	-19.3	-9.4	-13.8	-3.5	-6.3	-2.9	-9.5	-19.2	-12.9	-12.8	-5.7	-7.8
ResNet50	-4.8	-28.0	-12.6	-12.4	-10.6	-18.8	-5.0	-8.8	-5.3	-11.6	-23.2	-11.2	-12.0	-6.3	-10.5
Swin-B	-5.6	-17.0	-9.9	-10.5	-10.1	-12.3	-3.4	-5.3	-3.3	-8.7	-14.0	-12.1	-10.9	-10.3	-8.5
GCM	9														
DNN / Factor	B	G	IN	N	S										
ConvNext-B	0.30	-7.5	-7.7	-6.7	-7.1										
ResNet50	-0.15	-9.8	-12.5	-9.9	-9.3										
Swin-B	0.30	-6.6	-6.5	-6.3	-6.6										

Table A2. **Estimated** $\widehat{ACE}_{acc}(V\!:\!0\!\rightarrow\!1)$ (%) **calculated for each GCM and factor** Each value represents the expected change in accuracy as a result of increasing the corruption severity associated with the corresponding variable. Note how ACE changes for different nodes as a function of the DAG topology. Values close to 0 (or > 0) indicate higher robustness.

GCM	0					1					2				
DNN / Factor	G	IN	N	P	S	C	G	GN	IN	P	B	D	G	GN	N
ConvNext-B	-5.7	-6.6	-3.5	-8.2	-7.4	-21.8	-2.1	-4.2	-9.8	-1.9	-2.0	-8.7	-2.7	-2.6	-7.0
ResNet50	-7.1	-8.9	-4.7	-5.6	-12.2	-26.2	-3.7	-5.1	-12.1	-4.1	-1.7	-10.9	-3.9	-3.2	-14.1
Swin-B	-4.8	-6.1	-3.8	-10.3	-6.7	-16.6	-3.4	-3.9	-8.8	-7.9	-1.4	-9.0	-3.8	-2.4	-5.1
GCM	3					4					5				
DNN / Factor	D	G	IN	N	P	G	IN	P	S	SN	B	D	G	IN	P
ConvNext-B	-6.0	-3.0	-5.3	-2.0	-5.0	-4.5	-5.7	-12.8	-8.8	-4.2	-0.56	-5.5	-0.05	-4.0	-7.5
ResNet50	-9.3	-5.3	-8.5	-4.4	-0.17	-5.5	-7.9	-14.0	-11.0	-8.3	-0.75	-7.5	0.47	-6.2	-8.4
Swin-B	-5.5	-2.5	-5.2	-2.6	-7.5	-4.0	-4.6	-14.0	-7.8	-4.3	-0.64	-5.0	-0.73	-3.4	-6.6
GCM	6					7					8				
DNN / Factor	B	C	GN	N	S	D	GN	IN	N	S	C	GN	IN	P	S
ConvNext-B	-6.8	-29.7	-11.9	-18.8	-7.6	-13.9	-2.7	-6.8	-4.0	-10.3	-17.7	-10.0	-12.1	-6.7	-8.0
ResNet50	-2.9	-28.4	-11.5	-12.7	-10.6	-18.1	-4.2	-8.9	-4.3	-10.7	-22.5	-8.5	-12.1	-6.9	-11.1
Swin-B	-4.8	-18.2	-9.6	-10.5	-8.4	-12.0	-2.9	-5.2	-3.9	-8.0	-12.6	-9.3	-10.3	-11.1	-8.8
GCM	9														
DNN / Factor	B	G	IN	N	S										
ConvNext-B	0.20	-6.7	-6.6	-5.9	-6.9										
ResNet50	-0.33	-10.7	-11.5	-8.9	-10.1										
Swin-B	-0.82	-5.6	-5.1	-6.5	-5.8										

Table A3. **True** ACE_{acc} **for GCMs in common corruptions framework**. Each column represents the $ACE(V\!:\!0\!\rightarrow\!1)$ for a DAG with a single corruption variable and edge pointing to the corrupted image (as in the common corruptions framework of Figure 2a).

	Common corruptions framework									
DNN / Factor	B	C	D	G	GN	IN	N	P	S	SN
ConvNext-B	-4.4	-7.2	-14.4	-8.2	-9.0	-12.4	-10.0	-9.5	-5.8	-9.4
ResNet50	-4.7	-14.1	-19.6	-10.6	-16.7	-29.1	-19.0	-12.8	-11.8	-16.8
Swin-B	-4.4	-7.0	-15.6	-9.3	-8.8	-10.8	-9.8	-9.4	-6.4	-9.3



(a) GCM 0

(b) GCM 1

(c) GCM 2

(d) GCM 3

(e) GCM 4

(f) GCM 5

(g) GCM 6

(h) GCM 7

(i) GCM 8

(j) GCM 9

Figure A1. **DAGs for the randomly generated GCMs used to produce the data and results of Table A2.** Edges from each factor in $\mathcal{V}$ to X and between $\{X, Y, \hat{Y}, M\}$ are omitted for visual clarity.

Table B7. **Effect of DAG errors on ACE estimation for N_E edge errors.** Calculated as the deviation from the baseline estimation error (in $\%$) when the DAG is correctly specified. Close to 0 is best. (See Table B6 for baseline errors)

(a) Missing edges

DNN	ConvNext-B			ResNet50			Swin-B		
N_E	1	2	4	1	2	4	1	2	4
0	-0.10	-0.28	-0.21	-0.17	-0.27	-0.31	-0.06	-0.15	-0.15
1	-0.27	0.03	0.55	-0.14	0.06	0.38	-0.12	0.24	0.82
2	0.06	0.11	0.31	0.15	0.24	0.64	0.06	0.12	0.33
3	0.19	0.31	0.49	0.21	0.41	0.55	0.19	0.37	0.55
4	0.08	0.09	0.06	0.09	0.06	-0.04	0.09	0.07	0.00
5	-0.04	0.02	0.08	-0.05	0.16	0.32	-0.01	0.14	0.26
6	0.24	0.50	0.77	0.15	0.35	0.80	0.22	0.53	0.91
7	0.12	0.16	0.23	-0.08	-0.12	-0.02	0.02	0.08	0.01
8	0.29	0.64	1.6	0.21	0.44	0.83	0.29	0.60	1.3
9	0.01	0.06	0.46	0.08	0.22	0.64	0.05	0.06	0.39
Mean	0.06	0.17	0.44	0.04	0.16	0.38	0.07	0.21	0.44
Std	0.50	0.79	1.3	0.45	0.73	1.0	0.44	0.78	1.2

(b) Added edges

DNN	ConvNext-B			ResNet50			Swin-B		
N_E	1	2	4	1	2	4	1	2	4
0	-0.01	-0.02	-0.02	0.01	-0.01	-0.00	0.00	0.00	0.01
1	0.06	0.11	0.10	0.05	0.12	0.10	0.07	0.12	0.10
2	0.01	0.00	-0.07	0.04	0.06	-0.03	0.02	0.03	-0.03
3	0.02	0.04	0.04	0.03	0.04	-0.00	0.04	0.07	0.06
4	-0.00	0.01	0.01	-0.02	-0.04	-0.02	-0.03	-0.05	-0.03
5	-0.03	-0.02	-0.04	-0.02	-0.01	-0.07	-0.07	-0.05	-0.02
6	0.00	-0.03	-0.03	-0.01	-0.04	-0.03	0.01	-0.01	-0.01
7	0.12	0.11	0.14	0.03	0.01	0.04	0.07	0.06	0.09
8	0.06	0.07	0.17	0.04	0.04	0.17	0.01	0.01	0.11
9	0.03	0.05	0.04	0.05	0.06	0.02	0.06	0.07	0.05
Mean	0.03	0.03	0.04	0.02	0.02	0.02	0.02	0.03	0.03
Std	0.11	0.13	0.13	0.10	0.13	0.14	0.11	0.13	0.11

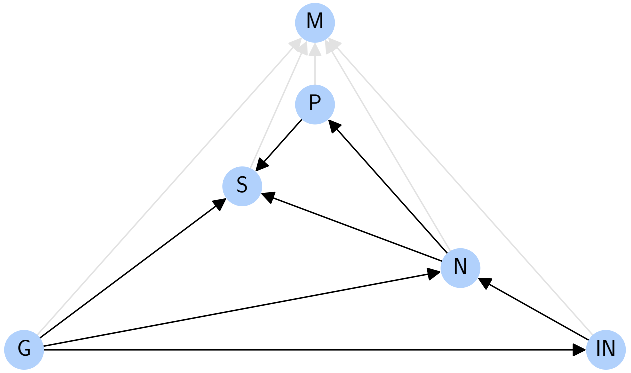
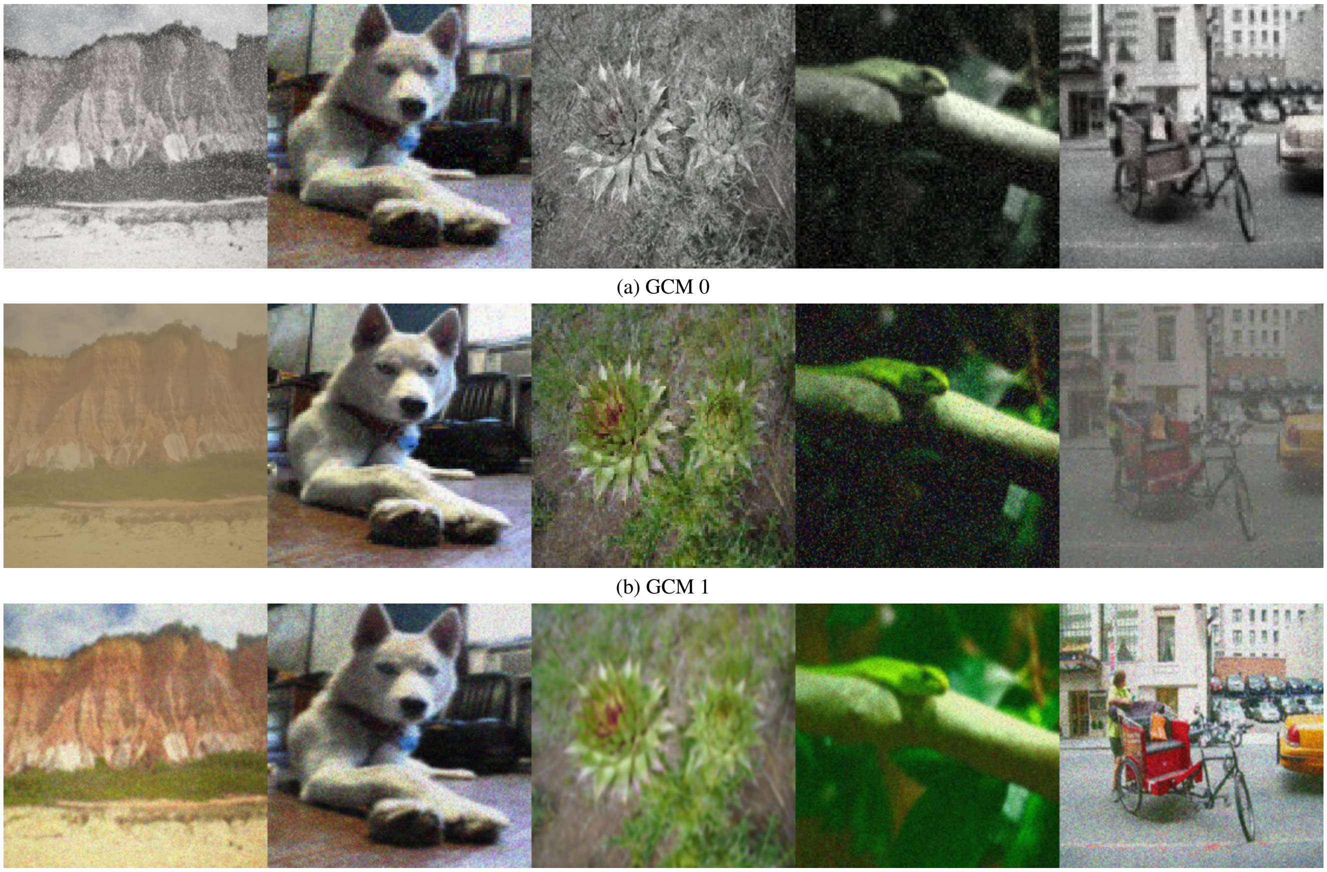


Figure C2. Causal DAG for GCM 0 including the node M corresponding to the correctness metric.

Table C8. Backdoor variables obtained via causal identification for GCM 0 in Figure C2

Variable	Adjustment Set
G	[]
IN	[G]
N	[G, IN]
P	[N]
S	[G, P, N]



(a) GCM 0

(b) GCM 1

(c) GCM 2

Figure D3. A random subset of images rendered according to the settings sampled from each corresponding GCM. The nature and severity of imaging conditions are diverse across GCMs illustrating how changes in the GCM DAG topology can have a measurable impact on the image quality. *Images best viewed digitally.*

where $\min_A$, $\max_A$ correspond to the minimum, maximum settings to be allowed in Blender respectively. For *decreasing* factors, the value $1 - V_A$ is computed first and then mapped to $[\min_A, \max_A]$. For *centered* corruptions, the value is first rescaled to $V'_A \in [\min_A, \max_A]$ and then adjusted according to $\frac{|V'_A - n_A|}{\max(|\min_A - n_A|, |\max_A - n_A|)}$ where n_A

Table A4. Mean Top-1 Accuracy for each of the GCM datasets.

GCM / DNN	ConvNext-B	ResNet50	Swin-B
0	60.9	39.8	59.7
1	53.4	33.4	55.8
2	68.2	48.1	67.2
3	62.4	42.5	60.9
4	58.2	35.8	57.4
5	66.0	49.4	64.2
6	51.6	34.2	59.4
7	60.2	35.3	60.9
8	48.8	27.9	51.4
9	64.7	41.2	64.6
Clean	83.7	76.1	83.2

is the nominal value that yields no/minimal effect of that attribute on the rendered image. Normalization settings are given in Tables E9 and E10.

F. CDRA for optical flow

For evaluating the robustness of optical flow methods, we also *render* a new variant of the Kubric MOVi-C benchmark dataset [2] using the same GCM DAG in Figure 5b and according to the process described in Appendix E. We evaluate three top performing baselines FlowNetC [1], PWCNet [3], and RAFT [4] and use the standard average Endpoint Error (EPE) for measuring overall performance and use CDRA to estimate ACE_{EPE}.

Results: The results in Table F11 show little difference in ACE_{EPE} and EPE between algorithms but large differences in ACE_{EPE} for each factor. Whereas average EPE is similar across all models, we observe that noise and defocus have much larger adverse effects on EPE relative to the other factors.

Discussion: These results indicate that CDRA exposes sensitivities of the optical flow models not observed when measuring only average performance on the dataset. These insights enable more targeted downstream robustness interventions in the DNN architecture design, training data collection, or optimization strategy to address these sensitivities.

G. Compute resources and requirements

All experiments can be executed using a single, locally-hosted NVIDIA A40 GPU with 48GB of memory. Data generation with both the compositing and rendering methods required a single A40 GPU as well. The causal effect estimation was conducted using a single laptop CPU.

The computational overhead for CDRA was minimal compared to evaluations in the common corruptions framework. The cost of evaluating task DNNs on domain/corrupted images is equivalent between both frameworks. In our experiments, a small amount of additional computation was necessary for estimating $\widehat{ACE}_M(V : v \to \tilde{v})$ which amounted to training a Random Forest Regressor on the evaluation outputs (which required a single laptop CPU).

References

[1] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 5

[2] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 5

[3] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 5

[4] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 5

Table A5. Mean Top-1 Accuracy for single factor GCMs (*i.e.* common corruptions framework - Fig. 2a) with corruption severity 1.

DNN / Factor	B	C	D	G	GN	IN	N	P	S	SN
ConvNext-B	79.3	76.5	69.3	75.5	74.7	71.3	73.7	74.2	77.9	74.3
ResNet50	71.4	62.1	56.5	65.5	59.4	47.0	57.1	63.3	64.3	59.3
Swin-B	78.7	76.1	67.5	73.9	74.4	72.3	73.4	73.8	76.8	73.9

Table B6. Baseline Δ_{ACE} $(\%)$ calculated for each GCM and averaged across all imaging factors in the GCM DAG.

GCM	ConvNext-B	ResNet50	Swin-B
0	1.0	0.83	0.70
1	1.3	0.84	0.87
2	0.84	1.1	0.85
3	0.71	0.62	0.74
4	0.89	0.54	0.73
5	0.57	0.36	0.45
6	1.2	0.81	0.97
7	0.70	0.68	0.51
8	1.2	0.97	1.2
9	0.61	0.85	0.89
Mean (all GCMs)	0.91	0.76	0.79
Std (all GCMs)	0.79	0.74	0.72

Table E9. Directed edge weights $\alpha_{A_i A_j}$ for each edge in the GCM of Section 5.3. Edge weights were sampled from $\mathcal{U}(-1, 1)$.

A_i	A_j	Weight
L	E	-0.223
L	D	-0.800
E	D	0.800
E	N	-0.322
D	N	-0.909

Table E10. Hyperparameters for each factor to sample corruption severities in Section 5.3

Factor	Render Setting	Corruption Type	Min	Max	Nominal	a_A	b_A	$f(\cdot)$	σ_A
L	Light Level	Centered	0.25	1.5	1	2	2	$(1 + tanh)/2$	1
E	Exposure	Centered	-2	2	0	3	3	$(1 + tanh)/2$	0.1
D	Defocus (via F-stop)	Decreasing	0.01	0.2	-	2	5	$(1 + tanh)/2$	0.1
N	Noise (via render cycles)	Decreasing	10	300	-	1	1	$(1 + tanh)/2$	0.1

Table F11. Comparison of ACE_{EPE} for multiple optical flow baseline algorithms.

	$ACE_{EPE}(\cdot)$				
DNN / Factor	C	E	F	L	EPE
FlowNetC	**18.7**	**-19.4**	**57.7**	**-1.19**	10.7
PWCNet	18.8	-19.5	58.0	-1.17	10.3
RAFT	18.8	-19.6	58.0	-1.18	**10.3**