Synthetic Designed Experiments for Diagnosing Vision Model Failures

Krisanu Sarkar
Indian Institute of Technology Bombay
Mumbai, India

Abstract

Current synthetic data pipelines for computer vision generate images without diagnosing what the downstream model actually needs. This open-loop paradigm treats synthetic data as cheap real data—randomly sampling the generator's output space and hoping for coverage of the model's failure modes. We argue that this fundamentally misuses synthetic data's unique property: controllable, independent variation of scene factors. Drawing on the statistical theory of Design of Experiments *(DoE)—a principled framework for efficiently probing complex systems that, to our knowledge, has not been applied to synthetic data generation for vision—we propose* ***Synthetic Designed Experiments for Representational Sufficiency (SDRS)****. SDRS treats the downstream model as a black-box system under investigation and the synthetic generator as an experimental apparatus. Using fractional factorial designs, SDRS efficiently audits a model's factor-sensitivity profile via ANOVA decomposition, classifying failures into two actionable types:* Type I gaps *(coverage failures on underrepresented factor levels) and* Type II gaps *(reliance on spurious nuisance dependencies). The audit then prescribes targeted synthetic data to address each gap type. We validate SDRS on three experiments: (1) a controlled diagnostic on dSprites with planted biases, where the audit correctly identifies both gap types and targeted data improves accuracy from 49.9% to 79.0%; (2) a dense segmentation task on procedural scenes, where the audit detects background-complexity shortcuts and targeted data improves mIoU from 0.948 to 0.998; and (3) an entanglement detection experiment showing that the ANOVA audit identifies cross-factor contamination in imperfect generators ($\Delta F = 4.7$ for the leaked factor). We further show that per-factor invariance penalties can transfer sensitivity between factors, identifying an open problem for representation-level correction. Code will be released.*

1. Introduction

The synthetic data pipeline for computer vision has converged on a standard recipe: generate large volumes of images using a controllable source (a 3D renderer [9, 19] or a conditioned diffusion model [15, 23]), optionally filter for quality, and add them to the training set alongside real data. This paradigm has demonstrated consistent gains across object detection [7], semantic segmentation [22], depth estimation [12], and human pose recovery [4].

Yet a fundamental question remains underexplored: *which* synthetic images should be generated? Current pipelines answer this question with heuristics—text prompts chosen by the researcher, domain randomization that uniformly samples all factor combinations, or difficulty scores derived from a pretrained classifier [21]. These approaches are **open-loop**: they do not use the downstream model's specific failure modes to guide generation. When the downstream model already handles 95% of the visual distribution competently, the vast majority of randomly generated synthetic images provide zero learning signal. This is a waste of computation and, more importantly, a missed opportunity.

We observe that controllable synthetic generators—whether 3D engines or layout-conditioned diffusion models—share a critical structural property: they expose *independent, named parameters* for scene factors such as lighting, viewpoint, material, and occlusion. This makes them functionally equivalent to the experimental apparatuses studied in the statistical theory of *Design of Experiments* (DoE) [5, 8], a framework developed precisely for the problem of efficiently understanding how a complex system responds to multiple controllable variables.

The connection is direct. In classical DoE, an experimenter varies input factors according to a structured plan (e.g., a fractional factorial design) and analyzes the system's output via ANOVA to decompose response variance into contributions from individual factors and their interactions. This structured approach is exponentially more sample-efficient than random sampling—a result established by Fisher in 1935 [8] and foundational to experimental science. To our knowledge, this connection has never been exploited in the synthetic data literature.

We propose **Synthetic Designed Experiments for Representational Sufficiency (SDRS)**, a framework that ap-

plies DoE principles to diagnose and address vision model failures. SDRS operates in four phases:

1. **Designed Experiment.** Generate a small, structured set of synthetic images by varying scene factors according to a fractional factorial design—requiring as few as 2^{k-p} images for k factors, rather than the full $\prod_i l_i$ combinations.
2. **Representational Audit.** Pass these images through the downstream model and perform ANOVA on the per-image task loss, decomposing loss variance into per-factor contributions. This produces a *factor-sensitivity profile* that reveals exactly which factors the model's predictions depend on.
3. **Gap Diagnosis.** Cross-reference the sensitivity profile with the known task structure. Nuisance factors with significant dependence are **Type II gaps** (spurious shortcuts). Factors with strong performance degradation on underrepresented or unseen levels are **Type I gaps** (coverage failures). This taxonomy is exhaustive: every factor falls into one of four quadrants.
4. **Targeted Prescription.** Generate synthetic data that specifically addresses each diagnosed gap: diverse samples along Type I factors to build missing capability, and matched counterfactual pairs along Type II factors to enable invariance regularization. Optionally, re-audit after training to verify convergence.

This framework makes three contributions:

(1) A principled diagnostic for synthetic data. The ANOVA-based audit provides a decomposed, per-factor assessment of what a model has and has not learned. This directly answers the workshop's call for "benchmark and evaluation methods for synthetic data" by turning the synthetic generator into a structured evaluation instrument.

(2) A unifying lens on existing approaches. We show that domain randomization [18], counterfactual data augmentation [13], and active learning [17] emerge as special cases of SDRS under specific assumptions in the SDRS framework, and that each corresponds to a suboptimal experimental design.

(3) Empirical validation and an identified open problem. Across three experiments—a controlled classification task on dSprites, a dense segmentation task on procedural scenes, and an entanglement detection study—we validate the diagnostic and demonstrate that data targeted by the audit dramatically outperforms no-synthetic baselines. We also identify a *sensitivity transfer* phenomenon: per-factor invariance penalties can suppress one shortcut while amplifying others, suggesting that holistic representation constraints are needed for the correction phase.

2. Related Work

SDRS sits at the intersection of synthetic-data generation, representation diagnostics, and experimental design.

Domain randomization and sim-to-real. Domain randomization (DR) [18, 19] improves robustness by sampling nuisance factors broadly, and has been successful in sim-to-real pipelines [11, 16]. However, DR is open-loop: it does not diagnose which factors actually drive downstream errors. SDRS adds a designed audit step that decomposes failure by factor before prescribing corrections.

Counterfactual and invariance-based methods. Counterfactual augmentation and invariance objectives [2, 13] aim to suppress spurious correlations. SDRS is complementary: it first identifies which nuisance factors are problematic, then targets those factors during correction. This reduces dependence on manual factor selection.

Uncertainty-guided generation and data selection. Active learning and uncertainty-guided generation [17, 21, 25] score sample informativeness at the sample level. SDRS instead provides *factor-level attribution*: it explains why uncertainty arises by decomposing error sensitivity across controllable factors.

Distillation and probing. Dataset distillation/condensation methods [6, 20, 24] optimize compact synthetic sets but are typically expensive and trajectory-dependent. Probing methods [1, 3, 10] analyze encoded information but are usually descriptive. SDRS uses ANOVA on task loss as a prescriptive diagnostic that directly determines what data to generate next.

3. Framework

We formalize SDRS as an audit-and-prescription loop for downstream model f_w and controllable generator G. The generator is parameterized by discrete factors $\mathbf{z} = (z_1, \ldots, z_k)$; factors may be semantic ($\mathbf{z}_S$) or nuisance ($\mathbf{z}_N$).

3.1. Phase 1: Designed Experiment

Instead of random sampling, we use a fractional factorial design. For k two-level factors, a Resolution IV design [5] uses 2^{k-p} runs (versus 2^k full factorial). Example: for k=5, a 2^{5-2}_{IV} plan uses 8 probe points while preserving unconfounded main effects.

Given designed settings $\mathcal{E} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(M)}\}$, we generate probe data

$$\mathcal{D}_{\text{probe}} = \{(x^{(j)}, y^{(j)}, \mathbf{z}^{(j)})\}_{j=1}^{M}, \qquad x^{(j)} = G(\mathbf{z}^{(j)}).$$

3.2. Phase 2: Representational Audit via ANOVA

For each probe sample we compute task loss $\ell^{(j)} = \mathcal{L}(f_w(x^{(j)}), y^{(j)})$ (scalarized per image for dense prediction). For each factor z_j, we run one-way ANOVA over

grouped losses and compute

$$F_j = \frac{\mathrm{MS}_{\text{between}}(z_j)}{\mathrm{MS}_{\text{within}}(z_j)}. \quad (1)$$

To control multiple comparisons, we apply Holm–Bonferroni correction over the k factor tests within each audit pass and report adjusted significance at α=0.05.

Why ANOVA on task loss rather than linear probing. Linear probes [1, 10] measure extractability from representations, whereas ANOVA on task loss measures prediction-level dependence, which is the operational quantity for failure diagnosis.

3.3. Phase 3: Gap Diagnosis

We use two explicit tests per factor:

- **Coverage test (Type I candidate):** stratified performance drop on underrepresented or unseen levels (estimated on an audit-validation split).
- **Shortcut test (Type II candidate):** nuisance-factor dependence detected by ANOVA ($z_j \in \mathbf{z}_N$ with Holm-adjusted significance).

Classification priority is deterministic: if the coverage test is positive, assign **Type I**; else if shortcut test is positive, assign **Type II**; else assign **Correct**. This resolves overlap cases (e.g., a nuisance factor that is both under-covered and sensitive) by prioritizing coverage correction first.

	Shortcut+	**Shortcut–**
Coverage+	**Type I**	**Type I**
Coverage–	**Type II**	Correct

3.4. Phase 4: Targeted Prescription

Type I correction (coverage restoration). For each Type I factor, generate diversity-focused synthetic data over the deficient levels while sampling remaining factors from the training distribution.

Type II correction (shortcut suppression). For each Type II factor, generate matched counterfactual pairs (x_a, x_b) with identical semantics and all nuisance factors fixed except z_j, then enforce representation invariance.

$$\mathcal{L}_{\text{inv}} = \frac{1}{|\mathcal{P}|} \sum_{(a,b)\in\mathcal{P}} \frac{1}{d_l} \left\| \Phi_w^{(l)}(x_a) - \Phi_w^{(l)}(x_b) \right\|_2^2, \quad (2)$$

where d_l is the number of elements in layer-l features (e.g., d_l=$H_l W_l C_l$ for dense maps), making regularization scale explicit across architectures.

Combined objective.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}}(\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{Type I}}) + \lambda\, \mathcal{L}_{\text{inv}}(\mathcal{P}_{\text{Type II}}), \quad (3)$$

with λ=0.5 in all experiments unless stated otherwise.

Verification. After each correction round, rerun the audit on the updated model. Stop when no practically significant Type I/II factors remain.

Relation to prior approaches. Domain randomization [18], counterfactual regularization [13], and active learning [17] can be viewed as partial instances of the pipeline: useful correction mechanisms without an explicit factor-level diagnosis stage.

4. Experiments

We validate SDRS in three settings: controlled diagnosis on dSprites, dense prediction on procedural scenes, and entanglement detection under an imperfect generator.

4.1. Experiment 1: Controlled Diagnostic Validation

Setup. We use dSprites [14], parameterized by `shape` (3), `scale` (6), `orientation` (40), `posX` (32), and `posY` (32). The task is shape classification, so `shape` is semantic and the remaining factors are nuisance.

Planted biases and splits. We construct a biased 30,000-image training set with two deficiencies: (i) a shortcut where shape is perfectly correlated with `posX`; and (ii) a coverage deficiency where only orientations $[0, 4]$ are observed. From a balanced pool, we create disjoint **audit-validation** and **final test** splits (5,000 each). The ANOVA audit and gap assignment are computed on the audit-validation split only; final accuracy is reported on the held-out test split.

Audit results. Table 1 identifies both planted issues. `posX` is significant despite being nuisance (F=2.07, adjusted p<0.001). `orientation` has high sensitivity (F=45.9) and, crucially, the largest stratified generalization drop on unseen levels, so it is assigned as a Type I coverage gap under the decision rule in Section 3.3.

Correction and baselines. SDRS uses 500 counterfactual pairs for `posX` (Type II) and 2,000 diverse-orientation samples (Type I), totaling 3,000 synthetic images. We fine-tune with Eq. (3) and compare against four baselines under the same data budget.

Table 1. **Experiment 1: ANOVA audit results (audit-validation split).** Holm-adjusted significance before and after correction.

	Before		After		
Factor	F	Sig.	F	Sig.	**Gap Type**
`shape`	219.7	***	8.3	***	(semantic)
`scale`	5.8	***	26.3	***	–
`orientation`	45.9	***	20.3	***	Type I
`posX`	2.1	***	1.0	n.s.	Type II
`posY`	1.5	*	1.5	*	–

Table 2. **Experiment 1: Held-out classification accuracy.** "Targeted Data" uses the same diagnosed samples with task loss only. "Random" and "DR" sample from the full balanced dSprites distribution and are reported as oracle references.

Method	**Accuracy**	**Data Source**
No synthetic data	0.499	–
SDRS (task + invariance)	0.751	SDRS-diagnosed
Targeted Data (task loss only)	0.790	SDRS-diagnosed
Random synthetic†	0.824	Oracle (balanced)
Domain randomization†	0.851	Oracle (balanced)

† Samples from full balanced data; see text.

Analysis. Training on diagnosed targeted data improves held-out accuracy from 49.9% to 79.0% (+29.1 points). The `posX` shortcut is removed (F: 2.1 $\rightarrow$ 1.0), and unseen-orientation accuracy improves substantially. Adding invariance loss lowers performance relative to task-loss-only training on the same diagnosed data (0.751 vs. 0.790), indicating objective competition in the current formulation. We also note `scale` rises from 5.8 to 26.3, an instance of sensitivity transfer discussed in Section 5.2.

4.2. Experiment 2: Dense Prediction on Procedural Scenes

Setup. We build a procedural generator for 128 $\times$ 128 RGB scenes with three objects and pixel-perfect segmentation masks (4 classes). Factors are `light_dir` (4), `light_int` (3), `bg_complex` (3), `obj_material` (3), `cam_angle` (3), and `occlusion` (3).

Planted biases and splits. The biased training set (500 scenes) fixes five factors and varies only `obj_material`, inducing nuisance shortcuts and an occlusion coverage gap. We create disjoint balanced **audit-validation** and **final test** splits (400 each). ANOVA and diagnosis are computed on the audit-validation split; mIoU is reported on the held-out test split.

Table 3. **Experiment 2: ANOVA audit and segmentation mIoU.** Audit computed on disjoint audit-validation data; significance is Holm-adjusted.

	Before		After		
Factor	F	Sig.	F	Sig.	**Gap**
`light_dir`	5.8	***	1.8	n.s.	Type II
`light_int`	10.3	***	97.6	***	↑
`bg_complex`	89.8	***	9.8	***	Type II
`obj_material`	1.5	n.s.	0.1	n.s.	–
`cam_angle`	2.2	n.s.	134.5	***	↑
`occlusion`	5.9	***	2.6	n.s.	Type I

Table 4. **Experiment 2: Held-out segmentation mIoU.** SDRS improves mIoU from 0.948 to 0.998. Task-loss-only training on the same diagnosed data achieves 0.9995.

Method	**mIoU**
No synthetic data	0.948
SDRS (task + invariance)	0.998
Targeted Data (task loss only)	0.9995
Random synthetic†	1.000
Domain randomization†	1.000

† Uniformly sampled factor space.

Audit results. Table 3 shows three Type II gaps and one Type I gap pre-correction. `bg_complex` dominates (F=89.8), and `occlusion` is the primary missing-coverage factor.

Correction and baselines. SDRS adds 100 targeted correction scenes and 200 counterfactual pairs. All baselines use the same correction budget of 100 additional scenes. All methods fine-tune from the biased model's pretrained weights.

Analysis. Diagnosed targeted data closes most of the gap to perfect segmentation (0.948 $\rightarrow$ 0.9995) while resolving the main audited dependencies. With invariance regularization, `cam_angle` and `light_int` increase sharply (2.2 $\rightarrow$ 134.5 and 10.3 $\rightarrow$ 97.6), indicating sensitivity transfer. Random/DR also reach 1.000 mIoU under the same 100-scene budget because this benchmark saturates quickly under uniform factor coverage. The diagnostic contribution remains: ANOVA identifies why the biased model fails (background shortcut and occlusion coverage failure), information that scalar mIoU alone cannot provide and that transfers to harder benchmarks where ceiling effects are less likely.

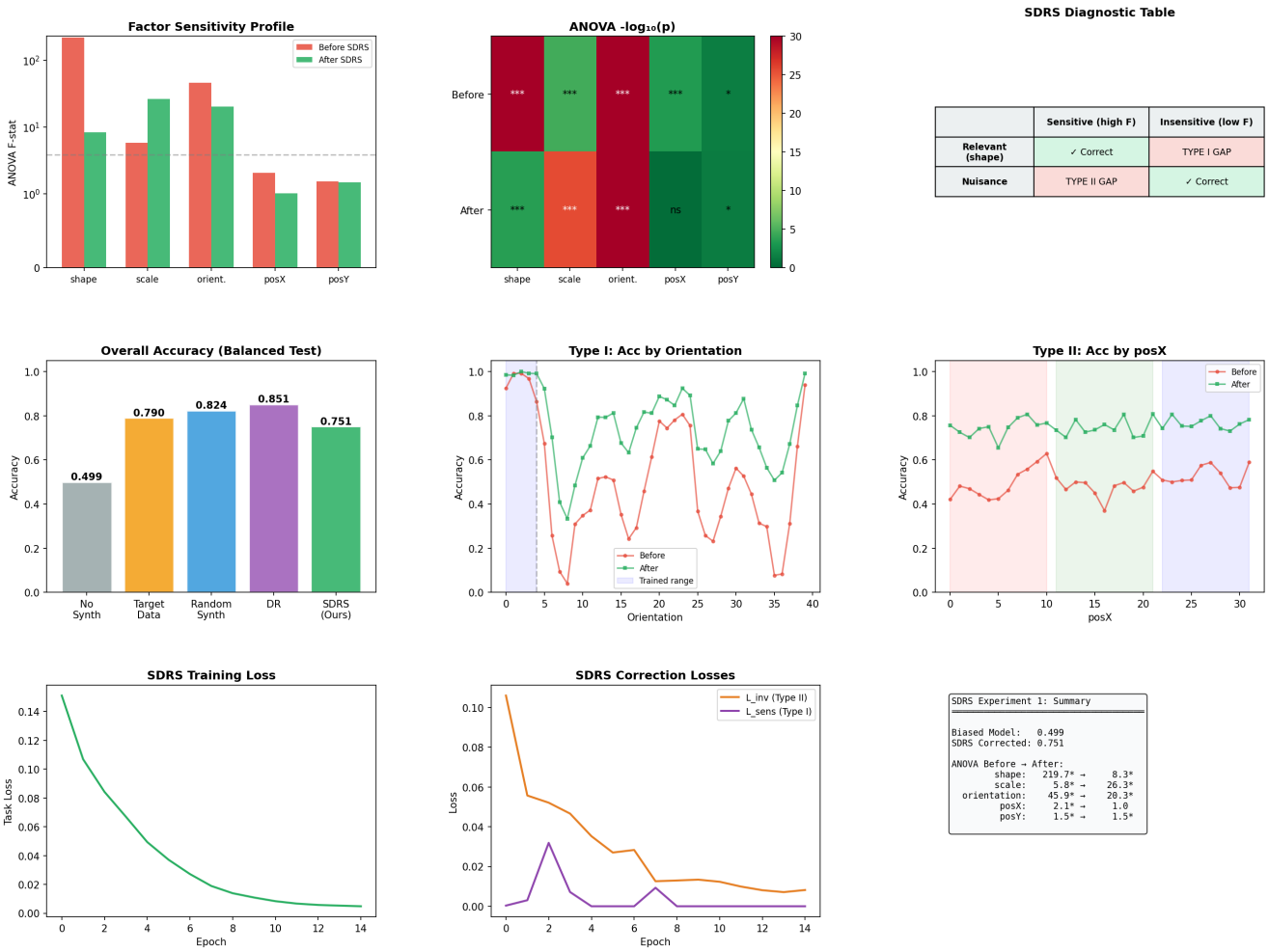

Figure 1. **Experiment 1: Controlled diagnostic validation on dSprites.** The audit identifies a `posX` shortcut (Type II) and an orientation coverage gap (Type I). After targeted correction, `posX` becomes non-significant and orientation sensitivity is reduced by 55.8%, with improved held-out accuracy. The in-figure diagnostic table illustrates the theoretical taxonomy; operational assignment follows the priority rule in Section 3.3.

4.3. Experiment 3: Detecting Generator Entanglement

Setup. We compare a perfect and an entangled procedural generator for 64×64 colored-shape images with factors `shape`, `color`, `size`, `style`, and `position`. In the entangled generator, `style` also changes object size (rough: +30%, sketchy: −21%).

Results and implications. Table 5 shows the expected shift under entanglement: `style` increases by +4.7 and `size` decreases by −7.9. This indicates that the ANOVA audit can also evaluate generator quality by flagging cross-factor contamination from empirical sensitivity profiles.

Table 5. **Experiment 3: Entanglement detection.** ANOVA F-statistics for perfect vs. entangled generators (audit-validation split).

Factor	F (Perfect)	F (Entangled)	ΔF
`shape`	8.7	4.6	−4.1
`color`	1.0	0.8	−0.1
`size`	11.4	3.6	−7.9
`style`	2.4	7.1	+4.7
`position`	3.2	3.2	0.0

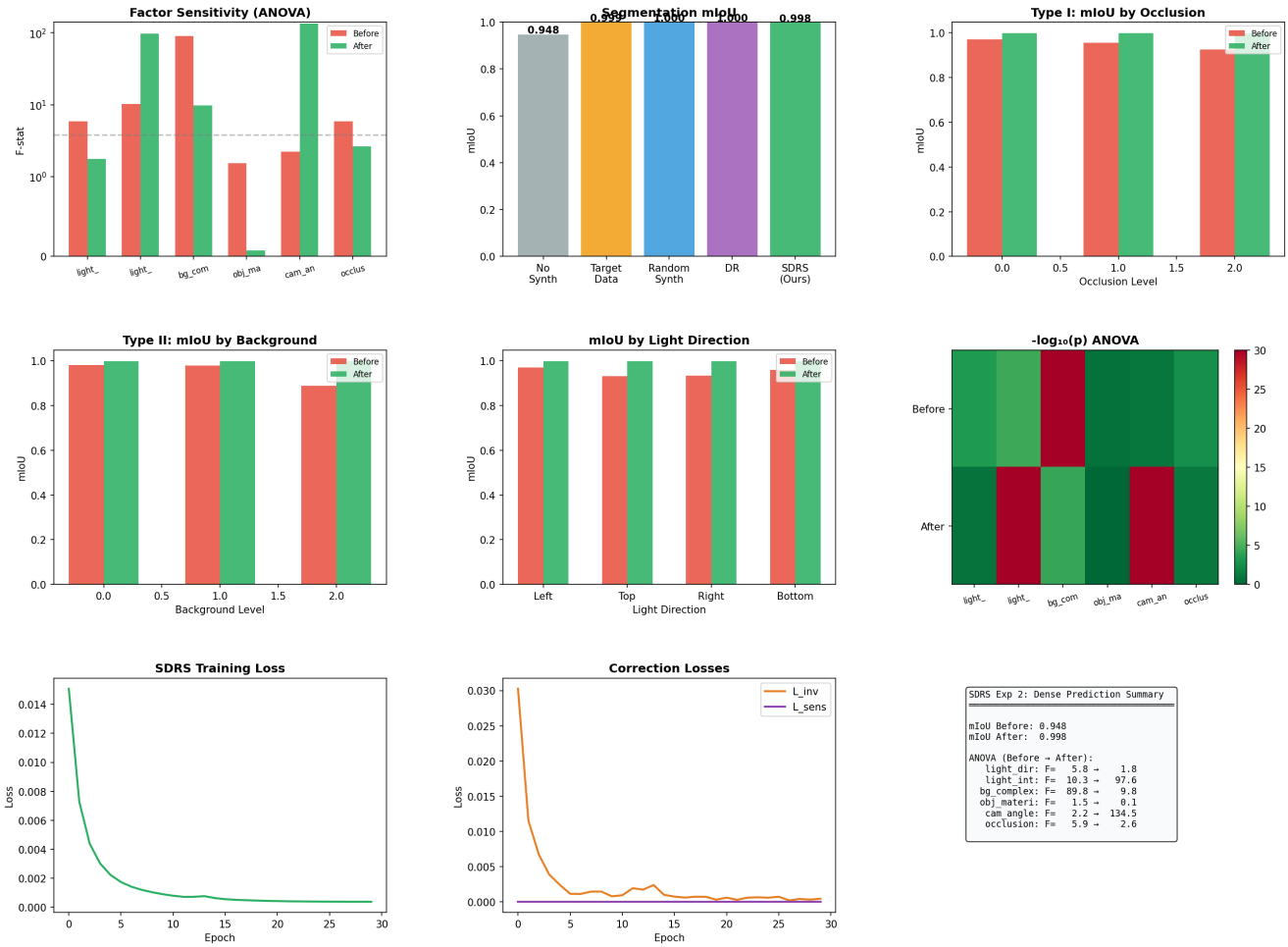

Figure 2. **Experiment 2: Dense prediction on procedural scenes.** SDRS reduces the dominant `bg_complex` shortcut (89.8 → 9.8), closes the occlusion gap, and improves held-out mIoU from 0.948 to 0.998. Task-loss-only training on the same diagnosed data reaches 0.9995, revealing sensitivity transfer under invariance regularization.

5. Discussion

5.1. The Diagnostic as the Primary Contribution

Across all three experiments, the ANOVA-based audit identifies the dominant vulnerabilities on disjoint audit-validation splits. In Experiment 1, it finds a `posX` shortcut and an orientation coverage failure; in Experiment 2, it identifies `bg_complex` as the dominant shortcut and `occlusion` as the primary missing-coverage factor; in Experiment 3, it detects style→size entanglement ($\Delta F = +4.7$) without ground-truth entanglement labels.

The practical value is decomposition: SDRS reports *which factors* drive failure and *which correction type* they require, rather than only aggregate scores such as accuracy or mIoU.

5.2. Sensitivity Transfer: An Identified Open Problem

In Experiment 2, the invariance penalty reduces dependence on targeted nuisance factors (`bg_complex`: 89.8 → 9.8; `light_dir`: 5.8 → 1.8) but increases sensitivity to non-targeted factors (`cam_angle`: 2.2 → 134.5; `light_int`: 10.3 → 97.6). We term this *sensitivity transfer*. A plausible mechanism is representational reallocation: suppressing some nuisance directions in $\Phi_w^{(l)}$ leaves capacity that is repurposed toward other nuisance cues.

This explains why the current correction objective does not consistently outperform task-loss-only training on

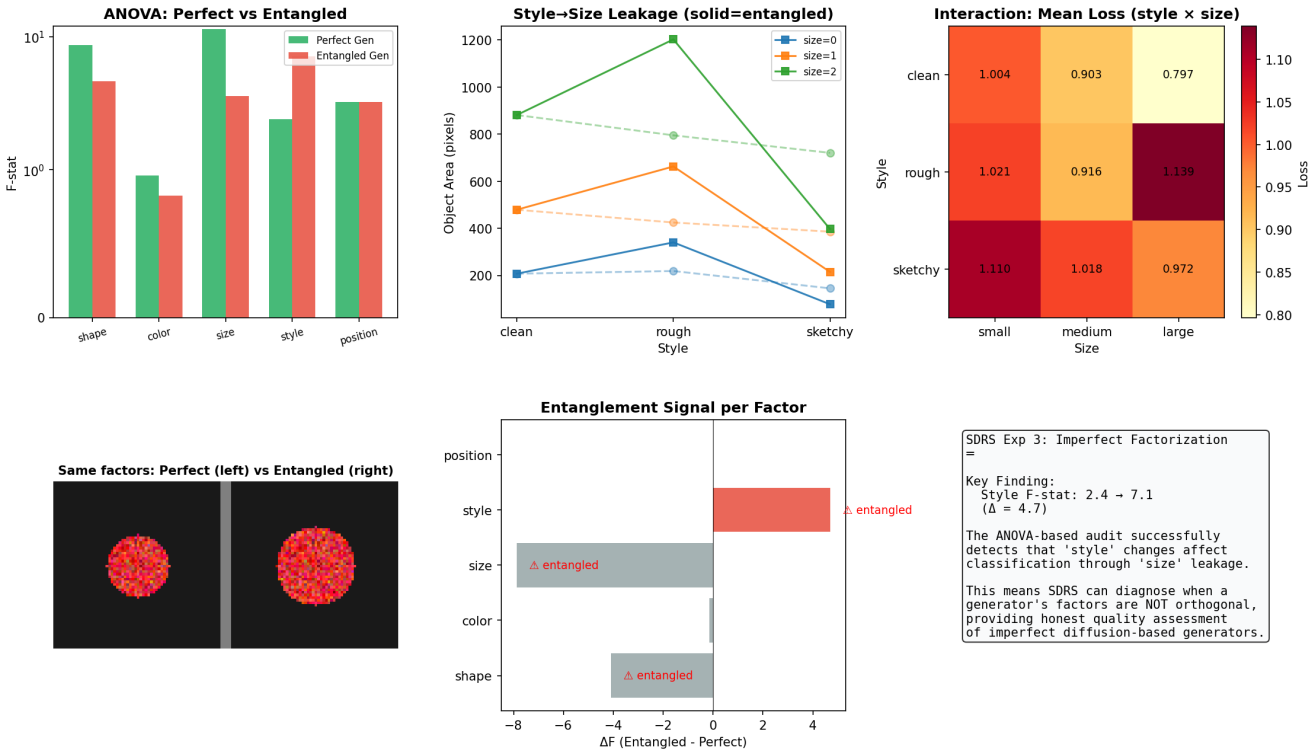

Figure 3. **Experiment 3: Detecting generator entanglement.** Comparing perfect and entangled generators, the audit detects style→size leakage: `style` rises from 2.4 to 7.1 while `size` drops from 11.4 to 3.6.

SDRS-diagnosed data (Table 2: 0.790 vs. 0.751; Table 4: 0.9995 vs. 0.998). In this version, the strongest contribution is the diagnostic stage; correction-loss design remains open.

5.3. Limitations

Factor coverage and interaction assumptions. SDRS audits only factors explicitly exposed by the generator. Unmodeled variation remains outside scope. The current one-way ANOVA focuses on marginal effects; interaction-heavy settings may require multi-way ANOVA/ANCOVA and larger designs.

Generator realism and factorization quality. Experiments use procedural generators with perfect (Experiments 1–2) or controlled (Experiment 3) orthogonality. Diffusion generators (e.g., ControlNet [23]) provide richer realism but weaker factor separation; broader validation on diffusion-generated data is still needed.

Correction loss robustness. As discussed in Section 5.2, per-factor invariance penalties can transfer sensitivity across nuisance factors. Improving correction losses beyond task-loss-only training on diagnosed data is a central next step.

6. Conclusion

We have presented SDRS, a framework that connects the statistical theory of Design of Experiments to synthetic data generation for computer vision. The core idea is to treat the downstream model as a system under investigation and the synthetic generator as an experimental apparatus, using structured factorial designs to efficiently probe the model's factor-sensitivity profile.

The ANOVA-based audit at the heart of SDRS provides a decomposed, per-factor diagnostic that identifies both spurious shortcuts (Type II gaps) and missing capabilities (Type I gaps). Across three experiments, the audit correctly identifies planted biases, tracks their evolution through correction, and detects cross-factor entanglement in imperfect generators. Data targeted by the diagnostic produces substantial performance gains over no-synthetic baselines (+29.1 pp in classification, +5.0 points mIoU in segmentation).

Our investigation also identified an open problem: per-factor invariance penalties can transfer sensitivity between nuisance factors rather than eliminating it, a phenomenon we term *sensitivity transfer*. This finding suggests that the correction phase of synthetic data pipelines requires more careful design than the field has so far recognized, and that the diagnostic framework we propose is a necessary first step—one must understand what is wrong before one can fix it.

We believe the central insight of this work—that synthetic data pipelines should be designed as structured experiments, not random sampling processes—provides a productive direction for the community, and we hope the SDRS diagnostic will serve as a practical tool for both evaluating models and assessing the quality of synthetic generators.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 2, 3

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulchanani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2

[3] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. 2

[4] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, 2023. 1

[5] George E.P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, 2nd edition, 2005. 1, 2

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *CVPR*, 2023. 2

[7] Kai Chen, Enze Luo, Shibo Xu, Zhengning Zhang, Jiayuan Jia, Zijin Fan, Zheng Liu, and Jing Shao. Geodiffusion: Text-prompted geometric control for object detection data generation. In *ICLR*, 2024. 1

[8] Ronald A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935. 1

[9] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, et al. Kubric: A scalable dataset generator. In *CVPR*, 2022. 1

[10] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP*, 2019. 2, 3

[11] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *CVPR*, 2019. 2

[12] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Mez, Tobias Dauber, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 1

[13] Chengzhi Mao, Augustine Cha, Amogh Gupta, Hao Wang, Junfeng Yang, and Carl Vondrick. Generative interventions for causal learning. In *CVPR*, 2021. 2, 3

[14] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 3

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[16] Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real single-image flight without a single real image. In *RSS*, 2017. 2

[17] Burr Settles. Active learning literature survey. *Computer Sciences Technical Report 1648, University of Wisconsin–Madison*, 2009. 2, 3

[18] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 2, 3

[19] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brober, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPR Workshops*, 2018. 1, 2

[20] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 2

[21] Zerun Wang, Chonghao Sui, Han Sun, Xiaojie Wang, Qiong-Hai Dai, and Yu-Chun Li. Difficulty-controlled diffusion model for effective synthetic dataset generation. *arXiv preprint arXiv:2411.18109*, 2024. 1, 2

[22] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023. 1

[23] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 1, 7

[24] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021. 2

[25] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017. 2

Synthetic Designed Experiments for Diagnosing Vision Model Failures

Supplementary Material

Table S6. Training hyperparameters across experiments.

	Exp 1	Exp 2	Exp 3
Learning rate	3×10^{-4}	1×10^{-3}	1×10^{-3}
Batch size	256	32	64
Epochs (biased)	15	30	15
Epochs (correction)	15	30	–
λ (Eq. (3))	0.5	0.5	–
Sensitivity margin	3.0	3.0	–
Inv. pairs per batch	32	32	–
Sens. pairs per batch	64	–	–

A. Model Architectures

Experiment 1: dSprites CNN. The downstream model is a four-layer CNN: `Conv2d(1,32,3,stride=2,pad=1)` → BN → ReLU → `Conv2d(32,64,3,2,1)` → BN → ReLU → `Conv2d(64,128,3,2,1)` → BN → ReLU → `Conv2d(128,256,3,2,1)` → BN → ReLU → `AdaptiveAvgPool2d(1)` → `Linear(256,128)` → ReLU → `Linear(128,3)`. Feature extraction for the invariance loss uses the 128-dimensional output of the penultimate linear layer. Total parameters: ∼430K.

Experiment 2: Small U-Net. We use a four-stage encoder-decoder with skip connections. Each encoder stage uses `Conv2d` → `BN` → `ReLU` → `Conv2d` → `BN` → `ReLU`, with channel progression $3 \to 32 \to 64 \to 128 \to 256$ in the encoder and $256 \to 128 \to 64 \to 32$ in the decoder. Upsampling uses bilinear interpolation followed by a convolutional block. The segmentation head is a 1×1 convolution mapping 32 channels to 4 classes. Feature extraction for the invariance loss uses the third encoder stage output ($32 \times 32 \times 128$). Total parameters: ∼1.9M.

Experiment 3: Tiny CNN. `Conv2d(3,16,3,2,1)` → ReLU → `Conv2d(16,32,3,2,1)` → ReLU → `Conv2d(32,64,3,2,1)` → ReLU → `AdaptiveAvgPool2d(1)` → `Linear(64,3)`. Total parameters: ∼25K.

B. Training Hyperparameters

All experiments use Adam with default momentum (β_1=0.9, β_2=0.999) and cosine-annealing learning-rate schedules. Table S6 summarizes the full settings.

C. Fractional Factorial Design Details

For Experiment 1 (k=5 factors), we use a 2^{5-2}_{IV} fractional factorial with generators D=AB and E=AC, yielding 8 runs. Each factor is mapped to two levels: low = first quartile of the factor range and high = last quartile. For example, `orientation` has 40 levels; low = $\{0, \ldots, 9\}$ and high = $\{30, \ldots, 39\}$.

In practice, we run the ANOVA audit on the balanced audit-validation split (5,000 images for Experiment 1 and 400 images for Experiment 2), rather than on a dedicated 8-point probe set. This provides higher statistical power while preserving the designed-intervention logic. Section 3.1 of the main paper therefore describes the *minimal* protocol when a balanced evaluation set is unavailable, whereas the experiments report results under the more favorable condition of balanced evaluation data.

D. Planted Bias Details

Experiment 1. The biased training set enforces:

- `shape`=0 (square) ⇒ `posX` $\in [0, 10]$
- `shape`=1 (ellipse) ⇒ `posX` $\in [11, 21]$
- `shape`=2 (heart) ⇒ `posX` $\in [22, 31]$
- `orientation` $\in [0, 4]$ (out of 40 levels)

Verification from training statistics: shape=0 has mean `posX`=5.0, shape=1 has mean `posX`=16.0, and shape=2 has mean `posX`=26.5. The biased model reaches 100% accuracy on the biased training set, confirming that the shortcut is trivially learnable.

Experiment 2. The biased training set fixes: `light_dir`=0 (frontal), `light_int`=1 (normal), `bg_complex`=0 (plain), `cam_angle`=0 (eye-level), and `occlusion`=0 (none). Only `obj_material` varies across its 3 levels.

E. Holm–Bonferroni Correction Procedure

Given k factor-wise ANOVA tests with ordered raw p-values $p_{(1)} \le p_{(2)} \le \cdots \le p_{(k)}$, the Holm–Bonferroni rule rejects $H_{0,(i)}$ if

$$p_{(i)} \le \frac{\alpha}{k - i + 1}, \tag{S4}$$

proceeding sequentially and stopping at the first non-rejection. For Experiment 1 (k=5, α=0.05), thresholds are 0.01, 0.0125, 0.0167, 0.025, and 0.05.

In Table 1 of the main paper, all factors except `posY` have $p < 0.001$, so they remain significant after adjustment.

Table S7. Complete ANOVA F-statistics before and after SDRS correction. Factors marked ↑ exhibit sensitivity transfer.

Experiment 1				
Factor	F_{before}	F_{after}	$\Delta\%$	Note
`shape`	219.7	8.3	-96.2	semantic
`scale`	5.8	26.3	$+355.9$	↑ transfer
`orient.`	45.9	20.3	-55.8	Type I corrected
`posX`	2.1	1.0	-50.8	Type II corrected
`posY`	1.5	1.5	-2.4	unchanged
Experiment 2				
Factor	F_{before}	F_{after}	$\Delta\%$	Note
`light_dir`	5.8	1.8	-69.9	Type II corrected
`light_int`	10.3	97.6	$+849.3$	↑ transfer
`bg_complex`	89.8	9.8	-89.1	Type II corrected
`obj_mat.`	1.5	0.1	-95.2	unchanged
`cam_angle`	2.2	134.5	$+5977.8$	↑ transfer
`occlusion`	5.9	2.6	-55.1	Type I corrected

`posY` has $p \approx 0.033$ and, as the largest p-value, is tested against 0.05; it remains significant.

F. Sensitivity Transfer: Additional Analysis

Table S7 reports complete before/after ANOVA profiles, including factors that exhibit sensitivity transfer (marked with ↑).

The transfer pattern is consistent across both experiments: factors not explicitly targeted by the invariance penalty can increase in sensitivity after correction. In Experiment 1, `scale` increases from 5.8 to 26.3. In Experiment 2, both `cam_angle` and `light_int` increase substantially. In contrast, targeted factors decrease as intended.

This behavior is consistent with a fixed-capacity representation hypothesis: suppressing targeted nuisance directions can free representational capacity that is then reallocated to other available cues, including non-targeted nuisance factors.

G. Entanglement Generator Specification

The entangled generator modifies rendered object size as a function of `style`:

- `style`=0 (clean): $\text{size}_{\text{rendered}} = \text{size}_{\text{base}}$
- `style`=1 (rough): $\text{size}_{\text{rendered}} = \text{size}_{\text{base}} \times (1 + \epsilon)$, with $\epsilon = 0.3$
- `style`=2 (sketchy): $\text{size}_{\text{rendered}} = \text{size}_{\text{base}} \times (1 - 0.7\epsilon)$, with $\epsilon = 0.3$

The perfect generator sets $\epsilon = 0$ for all styles. Base sizes are: small = 8 px, medium = 12 px, and large = 16 px (radius/half-side) on a 64×64 canvas.