MEMORIZE, ADAPT, IGNORE: DIAGNOSING ROBOT LEARNING MECHANISMS UNDER TRAINING DATA VARIATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training data variation, whether through designing a domain randomization (DR) scheme in simulation or curating demonstrations for imitation learning, is a primary lever for improving the robustness of robotic manipulation policies. Yet its underlying mechanisms remain poorly understood, and practitioners typically select randomization parameters through expensive trial and error. We investigate these mechanisms through a series of case studies, randomizing object size, color, and type as well as scene lighting and linguistic prompts across settings including pick-and-place RL in ManiSkill and fine-tuning of vision-language-action (VLA) models on LIBERO and RoboTwin. We examine both model behavior and internal representations, using the empirical neural tangent kernel (NTK) as our primary diagnostic tool. We show that the NTK distinguishes a shift in the internal learning mechanism from *memorizing* different situations with insufficient variation (e.g. learning what to do for a large cube, and what to do for a small cube) to *adapting* to the situation at hand with sufficient variation. An NTK-based signal-to-noise ratio also helps distinguish when policies have learned to *ignore* task-irrelevant factors (e.g. treating blue and red cubes identically, instead of learning a blue sub-policy and a red sub-policy). We use these diagnostics to develop practical guidance for designing DR schemes, selecting models, and detecting shortcut learning. We further compare different kinds of representations and validate our findings with real-world hardware experiments using ACT-based imitation learning.

1 INTRODUCTION

Data remains the primary bottleneck in modern robot learning. Generating high-fidelity real-world rollouts is prohibitively expensive, physical hardware data is notoriously scarce, and collecting expert human demonstrations is a painstakingly slow, manual process. To circumvent these physical limitations and bridge the notorious sim-to-real gap, practitioners deliberately inject variation into their training data. In simulation this typically takes the form of domain randomization (DR) (Tobin et al., 2017; Peng et al., 2018; Tremblay et al., 2018), which randomizes simulated parameters. In imitation learning, it comes from diversifying demonstrations or augmenting recorded observations. Training data variation is a primary lever for improving policy robustness. However, critical data-design decisions, such as which factors to vary, how much to vary them, and how to compose them, still rely almost entirely on raw human intuition or expensive, unguided empirical search. Roboticists are frequently trapped in trial-and-error loops, adjusting their data without a clear diagnostic compass.

Success rates on downstream tasks only reveal what a model has learned to do, failing to explain the underlying network mechanisms producing that learning. A policy might achieve high reward in simulation by engaging in fragile, brute-force memorization of discrete environments, by pathologically latching onto shortcut features, or by genuinely building a healthy, adaptive generalization model. These regimes demand fundamentally different interventions from the practitioner, but they are hard to distinguish with simple metrics like success rates. This leaves a major challenge in our ability to interpret, analyze, and optimize policies before deployment on real-world hardware.

To address this gap, this paper introduces a diagnostic framework based on representation kernels to isolate and track how training data variation shapes a model's internal representations during

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

training. A representation kernel measures how similar a network considers two inputs under some feature map, such as a layer's activations or the network's parameter gradients. Our default choice is the empirical Neural Tangent Kernel (eNTK), which directly describes how training on one sample propagates to others. We also compare the eNTK with last-hidden-layer and output kernels; while the latter two are much more computationally efficient, each has characteristic blind spots.

From these kernels we define various statistics that offer actionable insights for robotic policies trained with Reinforcement Learning (RL), Imitation Learning, and fine-tuning of Vision-Language-Action (VLA) models. Our geometric toolkit provides practitioners with the unique ability to explicitly distinguish whether a network is memorizing condition-specific variations, continuously adapting to task-relevant axes, or completely ignoring task-irrelevant distractors.

2 Background and Related Work

2.1 Representation Kernels

A representation kernel $K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ summarizes how a network represents inputs through a feature map ϕ; different choices of ϕ expose different parts of the network.

The most direct choice is a layer's activations, e.g., using ϕ as the last hidden layer $h(\mathbf{x})$ or the output $f(\mathbf{x})$, which reveal what features a network encodes or suppresses (Hermann & Lampinen, 2020). In robot learning, hidden states of trained policies encode symbolic task states (Lu et al., 2025), support behavior steering (Häon et al., 2025), and enable failure detection (Xu et al., 2025; Gu et al., 2026). Because robot actions are continuous and multi-dimensional, the output space is also informative (Agia et al., 2024). Related work links properties of pretrained representations to policy robustness (Burns et al., 2023; Jiang et al., 2025). Unlike these prior papers, we use representations to diagnose how a policy handles variation in its training data.

The Neural Tangent Kernel (Jacot et al., 2018) instead uses parameter gradients as features:

$$K_t(\mathbf{x}, \mathbf{x}') = \langle \nabla_\theta f(\theta_t, \mathbf{x}), \nabla_\theta f(\theta_t, \mathbf{x}') \rangle, \tag{1}$$

which governs how a gradient step on $\mathbf{x}$ changes the prediction at $\mathbf{x}'$ (see e.g. Ren, 2025). The kernel stays fixed in the lazy-training regime (Chizat et al., 2019), but even in the feature-learning regime of practical networks, the empirical NTK (eNTK) reflects the current representation and how it evolves (He & Su, 2020; Fort et al., 2020; Wei et al., 2022; Mohamadi et al., 2024; Ren, 2025).

Because evaluating the eNTK over all model parameters is expensive, we often restrict the Jacobian to low-rank adaptation (LoRA) modules (Hu et al., 2022) inserted into selected weight matrices. For a linear layer $z = Wh$ with LoRA update $W + sBA$, the standard initialization with random A_0 and $B_0 = 0$ gives tangent features proportional to $\delta(\mathbf{x}) \otimes (A_0 h(\mathbf{x}))$ for each output coordinate, where h is the layer input and δ is the backpropagated sensitivity. This maps the input factor of the full-weight feature $\delta(\mathbf{x}) \otimes h(\mathbf{x})$ into a low-dimensional space, yielding a cheaper sketch of the selected layers' kernel contributions. The restricted eNTK is also a diagnostic of the existing representation: because LoRA often adapts effectively from that representation, its tangent-feature similarities indicate how an update on one data point would affect another, even when no LoRA fine-tuning is actually performed.

2.2 Data Variation and Generalization in Robot Learning

A central goal in robot learning is generalization to conditions unseen in training, and the main lever for achieving it is variation in the training data. Domain Randomization (DR) serves as a primary paradigm for bridging the sim-to-real gap by exposing models to a diverse distribution of simulated environments (Tobin et al., 2017; Peng et al., 2018; Tremblay et al., 2018). This approach has been extensively validated across various tasks, including robotic grasping (Tobin et al., 2018), object detection (Horváth et al., 2023; Ren et al., 2019), dexterous in-hand manipulation (Akkaya et al., 2019; Handa et al., 2023), and bimanual manipulation (Chen et al., 2025). To move beyond uniform sampling, researchers have introduced structured and context-aware randomization (Prakash et al., 2019; Zakharov et al., 2019; Yue et al., 2019) and investigated the empirical limits of visual sim-to-real transfer (Alghonaim & Johns, 2021; Chen et al., 2022; Kim et al., 2023).

In imitation learning, variation comes from demonstration diversity, and a growing body of work studies how it shapes generalization. Data quality and composition strongly affect performance (Man-

dlekar et al., 2021; Belkhale et al., 2023; Hejna et al., 2024). Generalization differs across factors of variation (Xie et al., 2024; Pumacay et al., 2024) and can compose across them (Gao et al., 2024). At scale, environment and object diversity drive generalization more than demonstration count (Lin et al., 2025), yet not every kind of diversity helps (Saxena et al., 2025; Shi et al., 2025).

These studies characterize generalization through outcomes such as success rates. They show *whether* a policy generalizes across a factor, but not *how*: whether it memorizes, adapts to, or ignores that factor. We treat DR and demonstration diversity as instances of *training data variation* and diagnose these mechanisms using metrics on representation kernels.

3 METHODOLOGY

Our pipeline consists of three stages designed to isolate the effects of specific training data variation factors on a model's internal representations.

Stage 1: Controlled Training. We systematically vary one or two target factors while holding the distributions of all other factors fixed, training models across different DR configurations. Consider a robotic pick-and-place task as an example. Some factors are task-relevant (object shape and size, linguistic instructions), while others are task-irrelevant (object color or background lighting). When randomizing two factors simultaneously, we structure their joint distribution as either *confounded*, where the factors co-vary (for instance, a specific object color always pairs with a specific size), or *independent*, where all possible combinations of factors occur (a *factorial design*).

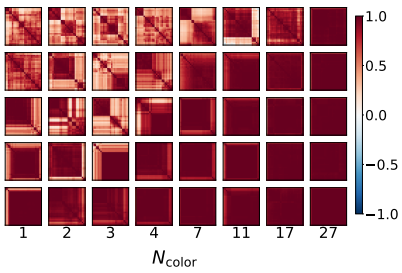
Stage 2: Probe Construction. To evaluate the models, we construct specialized probe sets that sweep through a range of values for a target factor while keeping all other input variables static. The first source of these probes is expert trajectories, obtained by taking standard expert demonstrations but re-rendering them with varying factors: for instance, changing the color of a cube while keeping all positions and actions identical. The second source is validiation frames, where we re-render static frames from the validation set with altered lighting or novel object and background appearances. By holding all other content strictly constant across these probes, any observed changes in the model can be directly attributed to the specific factor under study.

Stage 3: Representation Kernel Diagnostics. For each considered model checkpoint and probe set, we compute a representation kernel, as discussed in Section 2.1, on a subset of inputs. By default, we use the eNTK: it aggregates information across the entire network and characterizes how training on one input propagates to others. We will empirically compare to hidden-layer and output-layer kernels in Section 4. High values indicate that the model treats x and x' similarly during optimization; for instance, a gradient step changing the action on x will substantially affect the action on x'. We extract several statistics from this kernel to measure various effects in data variation.

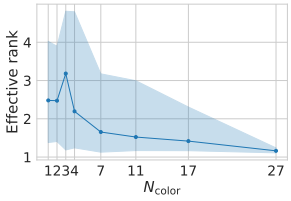
To develop these diagnostic methods, we will begin with a "subproblem" of a pick-and-place task: training a CNN to output the size of a square in an image. We will expand to robotic policies afterward, but it will be clearer to start with this warm-up. We will vary the number of colors considered for the square as $N_{\text{color}} \in \{1, 2, 3, 4, 7, 11, 17, 27\}$. For each value of N_{color}, we randomly sample 10 distinct color subsets to evaluate the performance variance under identical levels of data variation.

We construct 27 data probes representing different square colors, and compute the representation kernel matrix for each trained model configuration. Figure 1a shows these matrices. We can see that when the model is trained on only a few colors, the kernel matrix often exhibits a clear block structure, indicating that the model is treating these colors as isolated, distinct cases. That is, training on a blue square may barely change the model's behavior on red squares. We call this a *memorize* regime, as the model memorizes what to do for different situations, rather than generalizing across them. As the model is trained on more colors, this block structure flattens into an *adapt* regime. This shift is quantitatively captured in Figure 1b, where the **effective rank** (Roy & Vetterli, 2007) of the kernel matrices first increases as it is trained on more colors, then decreases to treat all colors similarly.

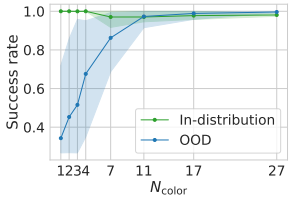
We compute both the in-distribution and out-of-distribution success rates across all N_{color} values, averaging the results across the 10 random color subsets. As shown in Figure 1c, while nearly all randomization levels achieve near-perfect in-distribution success rates, they diverge significantly in their OOD performance. Most notably, the surge in success at out-of-distribution prediction occurs around the same time as the kernel's effective rank flattens.



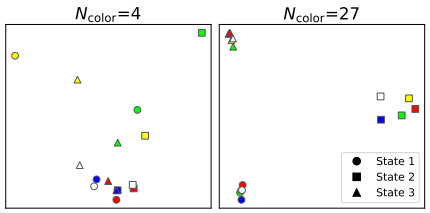
(a) Phase transition heatmap.



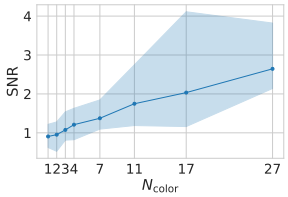
(b) Effective rank.



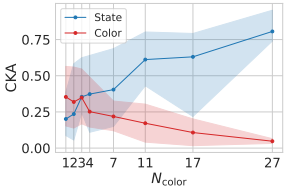
(c) Success rate.



(d) PCA adapt vs ignore.



(e) Signal-to-noise ratio.



(f) CKA

Figure 1: **Diagnostic metrics tracking representation kernel phase transitions.** As the number of randomized colors increases, the model's learning mechanism transitions from *memorizing* isolated environments (high effective rank, block-structured eNTK) to *adapting* across them, and finally to completely *ignoring* the task-irrelevant nuisance features (collapsing states via PCA, rising SNR, and state CKA alignment) to maximize OOD generalization.

To investigate this phenomenon, we construct cross-wise paired data probes between the task-irrelevant square colors and the task-relevant states. To better understand how the representation changes across the phase transition, we first visualize the probe representations using a two-dimensional PCA projection. Figure 1d shows the resulting projections for $N_{\text{color}} = 4$ and $N_{\text{color}} = 27$. When trained on only a few colors, probes are broadly separated in the feature space, including along the task-irrelevant color dimension. As the number of training colors increases, probes with the same task-relevant state but different colors become increasingly collocated, suggesting that the representation becomes less sensitive to color while preserving state-dependent structure.

For a more quantitative measurement, we build on distances in the feature space induced by the representation kernel: $d(x, x') = \|\phi(x) - \phi(x')\| = \sqrt{K(x,x) + K(x',x') - 2K(x,x')}$. To attribute changes in these representations to individual sources of variation, let $\mathcal{P}_a$ contain probe pairs that differ in factor a while matching in factor b, and let $\mathcal{P}_b$ contain the corresponding pairs that differ in b while matching in a. Letting $\bar{d}_a = \text{mean}_{(x,x')\in\mathcal{P}_a} d(x, x')$ and $\bar{d}_b = \text{mean}_{(x,x')\in\mathcal{P}_b} d(x, x')$, we define the *Factor Sensitivity Ratio* (FSR) as $\text{FSR}_{a:b} = \bar{d}_a/\bar{d}_b$. FSR asks how strongly the representation changes when varying one factor relative to another. When one factor s is task-relevant and another factor n is task-irrelevant, we call FSR by the familiar name of *Signal-to-Noise Ratio*: $\text{SNR} = \text{FSR}_{s:n} = \bar{d}_s/\bar{d}_n$. A larger SNR indicates that task-relevant variation produces larger changes in the representation relative to task-irrelevant variation. In our color experiment, s is the square size, and n the square color. As shown in Figure 1e, SNR increases with N_{color}, consistent with the representation becoming increasingly sensitive to size relative to color.

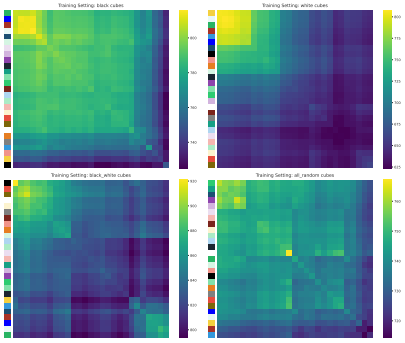
FSR and SNR provide a relative comparison between factors, but do not indicate which factor is responsible for a change in the ratio. To characterize the organization associated with each factor separately, we use Centered Kernel Alignment (CKA; Cortes et al., 2012; Kornblith et al., 2019). For a factor a, we measure the alignment of a label kernel $L_a(x, x') = \mathbf{1}[a(x) = a(x')]$ with K: $\text{CKA}(K, L_a) = \text{HSIC}(K, L_a)/\sqrt{\text{HSIC}(K,K)\,\text{HSIC}(L_a, L_a)} \in [0, 1]$, where $\text{HSIC}(K, L) \propto \text{tr}(KHLH)$ and $H = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$. High CKA indicates that the representation structure is more similar for examples with the same value of factor a. The state and color CKA curves in Figure 1f provide a factor-specific view of the same transition captured by the SNR in Figure 1e.

(a) **Predictive Validation of NTK Geometry.** Spearman Rank Correlation (ρ) and P-values between NTK cosine similarity and empirical success rates across 27 color candidates.

Training Cube	Spearman Rho (ρ)	p-Value
black	0.7065	0.0000
white	0.9165	0.0000
black_white	0.8189	0.0000
all_random	0.5712	0.0019

(b) **Metric evaluation across SNR, State CKA and Color CKA**. Bold values indicate superior performance.

Background	**Cube**	**SNR ↑**	**State CKA ↑**	**Color CKA ↓**
Pure	Pure	3.0404	**0.6620**	0.0576
Colorful	Pure	**3.4941**	0.6437	**0.0280**
Pure	Pure	2.5394	**0.7164**	0.0849
Pure	Colorful	**4.0933**	0.6692	**0.0167**



(c) **NTK similarity for cube-based training settings.** Hierarchical clustering reorders rows and columns for readability without changing the kernel. Settings: black cube only (top left), white cube only (top right), black and white cubes (bottom left), and random color cubes (bottom right).

Figure 2: CNN policies for a ManiSkill pick-cube task.

4 Experimental Analyses

RL Policies We now turn to a pick-cube task in the ManiSkill simulator (Tao et al., 2025), analyzing CNN policies trained via reinforcement learning across domain randomization intensities.

To investigate the internal learning mechanisms of the policy, we first establish the predictive power of the kernel geometry as a valid diagnostic proxy for empirical performance. Figure 2a evaluates the Spearman rank correlation between the kernel cosine similarity and empirical success rates across unseen evaluation colors. The high correlation across four training DR settings justifies using kernel statistics to reliably diagnose policy learning without requiring exhaustive evaluation rollouts.

We investigate the impact of visual diversity on representation structure by analyzing models trained under different cube-color configurations. Visualizing the resulting kernel matrices as heatmaps (Figure 2c) reveals a distinct structural phase transition from a *memorize* to an *adapt* regime. Insufficient DR yields prominent block-diagonal structures, demonstrating a memorization regime where the model treats different visual distributions as isolated, independent sub-tasks. Conversely, sufficient DR (e.g., the `random_color` setting) smooths out the kernel structure, indicating an *adapt*-style regime.

We also investigate whether policy learning suppresses task-irrelevant features by evaluating training performance under changing cube and background colors. For this analysis, we construct data probes cross-wise paired between scene states and colors. Quantitative alignment metrics, detailed in Table 2b, show that sufficient background and object DR lead to an *ignore* regime, which collapses irrelevant visual variance and lead to out-of-distribution (OOD) policy robustness.

Transformer Visuomotor Policies We train ACT policies for peg insertion under increasing table-texture variation. For tractability, we fix the CVAE latent z to its prior mean, as at inference, and compute the eNTK with respect to a LoRA adapter's parameters. Validation loss can be misleading for OOD robustness: a policy trained from scratch can overfit the in-distribution textures while achieving low validation loss. Indeed, the policy trained without variation attains the lowest validation loss, but the *worst* OOD success (Table 1a). In contrast, both SNR diagnostics correctly track OOD success, from 0.05 to 0.43.

We further vary two factors, texture and color, either individually or jointly, and evaluate on corresponding single-factor and joint shifts (Table 1b). Single-factor variation mainly improves generalization to that factor, whereas joint variation is needed for joint shifts, consistent with prior work (Gao

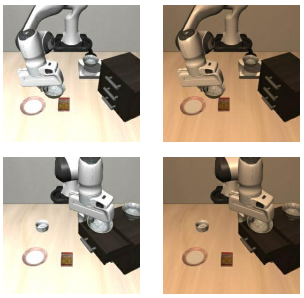
DR	Val. loss↓	SNR↑ act.	SNR↑ eNTK	SNR↑ OOD↑
none	**0.098**	0.37	0.54	0.05
mid	0.106	0.59	0.74	0.29
high	0.121	**0.79**	**0.89**	**0.43**

(a) Single factor (table texture).

DR	eNTK SNR (probes)↑ tex.	col.	joint	OOD success↑ tex.	col.	joint
none	0.45	1.19	0.42	0.05	0.35	0.00
color	0.45	**1.60**	0.51	0.08	**0.81**	0.04
texture	**0.62**	1.39	0.60	**0.43**	0.39	0.24
joint	0.56	**1.60**	**0.65**	0.31	0.49	**0.35**

(b) Two factors (texture, color), varied separately or jointly.

Table 1: **eNTK SNR tracks OOD success for ACT policies (peg insertion).** (a) The no-DR policy has the lowest validation loss but the worst OOD success; both SNRs increase with DR. (b) Joint-probe SNR orders all four training conditions by joint OOD success.



(a) Lighting perturbations.

Model	eNTK SNR	OOD Success
OpenVLA	2.802 ± 0.176	8.1%
RIPT-VLA	3.511 ± 0.195	88.4%
OpenVLA-OFT	3.534 ± 0.156	88.7%

(b) SNR (mean $\pm$ std dev) indicates that `OpenVLA` is the most lighting-sensitive, agreeing with its empirical behavior on the LIBERO-plus benchmark (Fei et al., 2025).

Figure 3: OpenVLA models evaluated on LIBERO under lighting perturbations.

et al., 2024). SNR on joint probes correctly orders all four training conditions by joint OOD success, showing that the diagnostic also captures the effect of training-data composition.

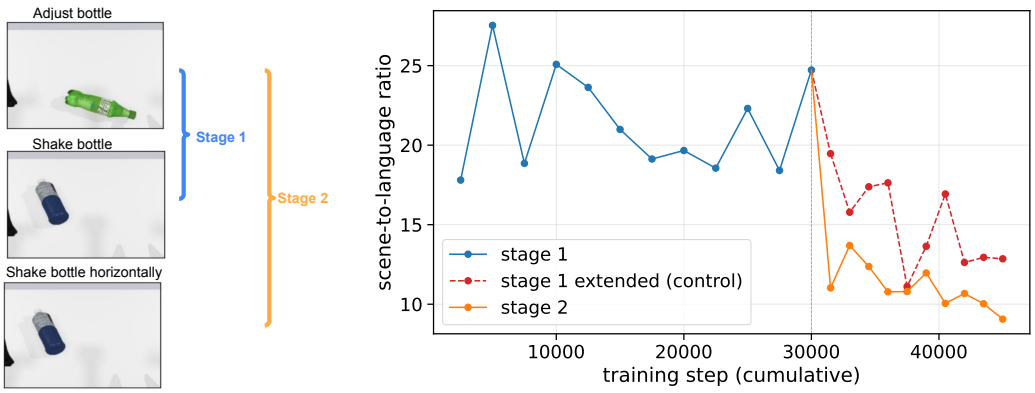
Scaling to Vision-Language-Action Policies We next apply our diagnostics to large Vision-Language-Action (VLA) policies on the LIBERO and RoboTwin benchmarks, computing the eNTK over parameter-efficient LoRA adapters as in Section 2.1. At this scale, the same FSR statistic supports two practical uses: ranking models by robustness and detecting shortcut learning, both without evaluation rollouts.

As we saw that randomizing task-irrelevant factors increases SNR and improves OOD performance, we use this insight we use this insight for zero-shot model selection. We treat lighting as a task-irrelevant nuisance and task state as the task-relevant signal, construct lighting-perturbation probes (Figure 3a), and compute $\mathrm{SNR} = \mathrm{FSR}_{\mathrm{state:lighting}} = \bar{d}_{\mathrm{state}}/\bar{d}_{\mathrm{lighting}}$ for the original OpenVLA (Kim et al., 2025b), the fine-tuned OpenVLA-OFT (Kim et al., 2025a) and RIPT-VLA Tan et al. (2025). A higher SNR means the representation is more sensitive to state than to lighting. The SNR ordering of the models matches their empirical OOD lighting robustness on the LIBERO-plus benchmark (Fei et al., 2025).

Shortcut learning (Xing et al., 2025) occurs when a policy relies on task-irrelevant cues rather than task-relevant information. To detect visual shortcut learning without task-level evaluation, we apply the FSR to compare representation sensitivity to scene and language variation. We evaluate OpenVLA-OFT on three RoboTwin tasks (Chen et al., 2025): adjust bottle, shake bottle, and shake bottle horizontally (Fig. 4a).

Stage 1 trains on the first two tasks, where each language instruction is perfectly confounded with a particular scene configuration. The model can therefore solve the training tasks by relying only on visual cues. We measure this using the scene-to-language ratio, $\mathrm{FSR}_{\mathrm{scene:language}}$.

Stage 2 introduces the third task, which breaks this visual–textual correlation and requires language to distinguish otherwise similar visual inputs. As shown in Fig. 4b, the FSR decreases after introducing the third task, indicating reduced relative sensitivity to scene variation and increased sensitivity



(a) Training setting illustration.

(b) Scene to language ratio for two training stages.

Figure 4: **Shortcut Learning Detection via Factor Sensitivity Ratio.** (a) Training setting illustrating the visual–textual confound in Stage 1 and its resolution in Stage 2. (b) Scene-to-language Factor Sensitivity Ratio (FSR) across training, computed as the representation change induced by scene variation relative to that induced by language variation. The decrease in FSR after introducing the third task indicates reduced relative sensitivity to scene variation and increased sensitivity to language variation.

to language. This drop is not simply a consequence of longer training. A control that continues first-phase training for the same 15k steps stays at 15.0, and the second phase is lower at all 10 step-matched checkpoints. FSR provides a representation-level signal of reduced visual shortcut reliance, without requiring task-level evaluation rollouts.

Independent Variation Beats Confounded Variation at Equal Data Budget Confounded variation does not necessarily cause shortcut learning, but it can introduce performance penalties. In Fig. 5b, erank transition on toy environment (Sec. 3) provides initial intuition: confounded variation remains more *memorize*-like, whereas independent variation transitions toward *adapt*.

We study this question in RoboTwin Place Object Scale task, where the policy places objects on a scale. We vary object shape, which is task-relevant, and texture, which is task-irrelevant. For each category, objects have similar but non-identical shapes. We compare *confounded variation*, which fixes one object-texture pair per category, with *independent variation*, which samples multiple objects with different textures within each category (Figure 5a). Both settings use the same data budget, i.e., the same number of training episodes and therefore the same data-collection effort. On novel object categories, independent variation achieves lower action error, while the two conditions are indistinguishable in-distribution (Figure 5c).

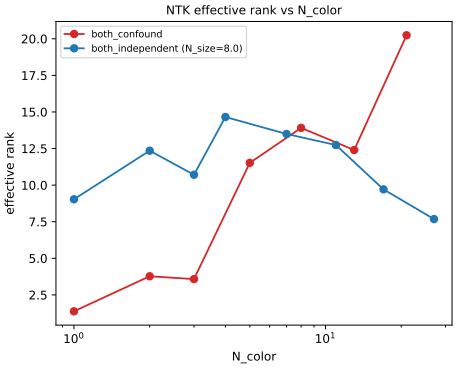
Representation Kernel Comparison We previously argued that the eNTK is a reasonable default representation kernel, rather than universally superior to activation-based alternatives. We compare it with the last-hidden-layer and output kernels on two diagnostics where representation choice matters. Since metric magnitudes are not directly comparable across representation spaces, we focus on qualitative trends within each representation.

We repeat the shortcut-detection study on 80 grasp-frame probes. For this experiment, we use the scene-to-language Factor Sensitivity Ratio (FSR), $\bar{d}_{\text{scene}}/\bar{d}_{\text{lang}}$, to quantify the relative sensitivity of each representation to the two factors. Both eNTK and last-hidden representations detect the alleviated shortcut at all 10 matched checkpoints ($p = 0.002$), whereas the output kernel does not ($4/10$, $p = 0.16$; Table 2a). At the grasp frame, both instructions require the same action, so the output representation collapses the language conditions. Thus, output-space probes cannot reveal reliance on a factor that does not yet affect the action.

We next use a bimanual task in which the two arms share all layers except their final heads. The last-hidden representation is therefore identical for both arms and cannot capture the arm-specific decoupling introduced by the heads. Here, each arm has a natural task-relevant factor (its own-peg



(a) Data variation design. Confounded variation ties shape and texture, whereas independent variation varies texture across objects with similar shapes within a category.



(b) Effective rank during training on toy environment. Independent variation shows an initial rise followed by a decline, marking a transition toward an *adapt*-style regime, while confounded variation keeps rising.

Evaluation	Confounded	Independent	Δ
In-distribution	0.0605	0.0587	-0.0018
Novel category	0.0846	**0.0755**	-0.0091

(c) Mean action error on evaluation categories. Both conditions use the same number of training episodes. Independent variation achieves lower error on novel categories (0.0755 vs. 0.0846), while in-distribution performance is similar (0.0587 vs. 0.0605).

Figure 5: **Independent variation improves generalization at equal data budget.**

pose) and a task-irrelevant factor (the partner-peg pose), so we use the signal-to-noise ratio (SNR), which is the corresponding special case of FSR. For each arm, the signal distance is placed in the numerator, with the ratio inverted when necessary so that $\mathrm{SNR} \geq 1$. The last-hidden SNR therefore remains near 1 because the two arms have identical last-hidden representations, while the output kernel detects the head-specific decoupling directly, with SNR increasing from about 2 to 48 for both arms. The eNTK captures the same trend more weakly (Table 2b).

The three representations have complementary blind spots: output space can miss distinctions that do not yet affect the action, while the last hidden layer can miss distinctions introduced in the head. The eNTK aggregates information across layers and has a direct physical interpretation through parameter-update effects, making it a reasonable default diagnostic representation in this paper. In practice, however, activation and output representations are substantially cheaper to compute and can be used within the same diagnostic framework. The representation can therefore be chosen case by case according to which aspects of the data it preserves.

Hardware Validation We validate the factor-conditioned kernel diagnostics on a hardware cube pick-and-place task using an ACT policy (Zhao et al., 2023). Lightness DR increases the training SNR from 2.5 to 5.5, indicating greater relative sensitivity to task-relevant state than to lighting variation (Figure 6a). This improvement is reflected in real-world robustness: success increases from $1/10$ to $8/10$ under bright lighting and from $1/10$ to $6/10$ under dark lighting, while in-distribution success increases from $6/10$ to $10/10$. Thus, the diagnostic tracks the improvement in visual robustness on hardware; full evaluation details are provided in Appendix A.1.

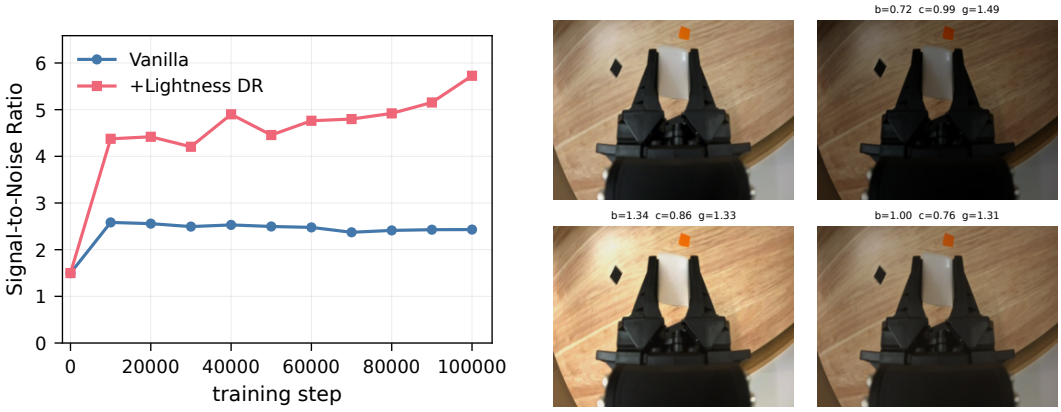
space	Stage 1	Stage 2	control	S2 < ctrl
eNTK	21.4	11.0	15.0	10/10
last hidden	8.5	7.0	9.5	10/10
output	148	136	127	4/10

(a) Scene-to-language FSR $\bar{d}_{\text{scene}}/\bar{d}_{\text{lang}}$ on 80 grasp-frame probes (lower = more reliance on language).

		training epoch			
arm	space	0	100	1900	trend
A vs B	output	1.9	15.0	47.8	↑
	last hidden	1.05	0.99	1.02	⇓
	eNTK (all)	1.02	1.07	1.14	↑
B vs A	output	1.4	15.8	47.7	↑
	last hidden	0.95	1.01	0.98	⇑
	eNTK (all)	0.97	1.08	1.12	↑

(b) Own-peg vs. partner-peg SNR in bimanual peg picking (CNN-MLP, 10×10 pose grid, 3 seeds).

Table 2: **Representation kernels have complementary blind spots.** (a) The output kernel misses the shortcut because both instructions require the same grasp-frame action. (b) The arms share all layers except their final heads, so the last-hidden kernel cannot see arm-specific structure.



(a) Signal-to-Noise ratio over training.

(b) Lightness DR samples.

Figure 6: Lightness domain randomization increases the SNR from roughly 2.5 to 5.5, indicating features become more separable across the augmented inputs. Samples are annotated with brightness $b \in [0.6, 1.4]$, contrast $c \in [0.7, 1.3]$, and gamma $g \in [0.7, 1.5]$, applied per-sample in that order.

5 Conclusion

We present a representation-kernel framework for diagnosing how robot policies respond to training-data variation. Across RL, imitation learning, and VLA models, our diagnostics distinguish three learning regimes: memorizing isolated conditions, adapting across task-relevant variation, and ignoring task-irrelevant variation. Kernel-based SNR and factor sensitivity provide signals for assessing OOD robustness and shortcut learning without relying solely on task-level evaluation.

Our results show that training-data design systematically shapes these learning regimes. Increasing relevant variation can drive policies from memorization toward adaptation, while variation in task-irrelevant factors promotes invariance. Independently varying factors also improves generalization over confounded variation at the same data budget. These patterns extend across model classes and are reflected in physical hardware performance, supporting representation kernels as practical diagnostics for data design and model analysis.

The framework has two main limitations. It requires probe sets defined around known factors and therefore cannot discover unknown sources of variation. Moreover, kernel metric magnitudes are not directly comparable across kernels, so our conclusions rely on qualitative trends and within-kernel comparisons. Overall, the framework provides a practical lens for understanding how training data shapes robot learning, complementing task-level evaluation and trial-and-error data design.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods. We have not used generative AI tools to generate synthetic data sets, clean and reformat dataset, and provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper, summarize or analyse existing literature, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, propose a title or keywords for a research paper. We have reviewed all AI-assisted work. We did manual literature survey to avoid potential plagiarism and checked the math and text generated by AI to ensure the correctness. We read the idea, hypothesis and result analysis generated by AI for reference rather than adopt directly. For AI-generated code, we manually reviewed and verified the math and crucial part, and tested the pipeline and infrastructure implementation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=yqLFb0RnDW.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Raghad Alghonaim and Edward Johns. Benchmarking domain randomisation for visual sim-to-real transfer. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. *Advances in neural information processing systems*, 36:80375–80395, 2023.

Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *ArXiv*, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025. URL https://arxiv.org/abs/2506.18088.

Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations*, 2022.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5850–5861. Curran Associates, Inc., 2020.

Jensen Gao, Annie Xie, Ted Xiao, Chelsea Finn, and Dorsa Sadigh. Efficient data collection for robotic manipulation via compositional generalization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40041–40076, 2026.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, and Yashraj Narang. DeXtreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5977–5984, May 2023. doi: 10.1109/ICRA48891.2023.10160216.

Bear Häon, Kaylene Caswell Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=YvsUD8C9QS.

Hangfeng He and Weijie Su. The local elasticity of neural networks. In *International Conference on Learning Representations*, 2020.

Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Remix: Optimizing data mixtures for large scale imitation learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=fIj88Tn3fc.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33:9995–10006, 2020.

Dániel Horváth, Gábor Erdős, Zoltán Istenes, Tomáš Horváth, and Sándor Földi. Object detection using sim2real domain randomization for robotic applications. *IEEE Transactions on Robotics*, 39 (2):1225–1243, 2023. doi: 10.1109/TRO.2022.3207619.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yTEwmr1TJb.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Texture learning domain randomization for domain generalized segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 677–687, October 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 54877–54910, 2025.

Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture. *2025 IEEE International Conference on AI and Data Analytics (ICAD)*, pp. 1–8, 2025. URL https://api.semanticscholar.org/CorpusID:276235652.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? A theoretical analysis on grokking modular addition. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35934–35967. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/mohamadi24a.html.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810. IEEE, May 2018. doi: 10.1109/icra.2018.8460528. URL http://dx.doi.org/10.1109/ICRA.2018.8460528.

Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255, 2019. doi: 10.1109/ICRA.2019.8794443.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The COLOSSEUM: A benchmark for evaluating generalization for robotic manipulation. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024. URL https://openreview.net/forum?id=aiSZqS5wkB.

Xinyi Ren, Jianlan Luo, Eugen SolowjoW, Juan Aparicio Ojea, Abhishek Gupta, Aviv Tamar, and Pieter Abbeel. Domain randomization for active pose estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7228–7234. IEEE Press, 2019. doi: 10.1109/ICRA.2019.8794126. URL https://doi.org/10.1109/ICRA.2019.8794126.

Yi (Joshua) Ren. *Learning dynamics of deep learning – force analysis of deep neural networks*. PhD thesis, University of British Columbia, 2025. URL https://open.library.ubc.ca/collections/ubctheses/24/items/1.0450240.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.

Vaibhav Saxena, Matthew Bronars, Nadun Ranawaka Arachchige, Kuancheng Wang, Woo Chul Shin, Soroush Nasiriany, Ajay Mandlekar, and Danfei Xu. What matters in learning from large-scale datasets for robot manipulation. International Conference on Learning Representations 2025, 2025.

Modi Shi, Li Chen, Jin Chen, Yuxiang Lu, Chiming Liu, Guanghui Ren, Ping Luo, Di Huang, Maoqing Yao, and Hongyang Li. Is diversity all you need for scalable robotic manipulation? *arXiv preprint arXiv:2507.06219*, 2025.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models, 2025. URL https://arxiv.org/abs/2505.17016.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.021.

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, Sep. 2017. ISSN 2153-0866.

Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3482–3489, 2018. doi: 10.1109/IROS.2018.8593933.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Alexander Wei, Wei Hu, and Jacob Steinhardt. More than a toy: Random matrix models predict how real-world neural representations generalize. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23549–23588. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wei22a.html.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2024.

Youguang Xing, Xu Luo, Junlin Xie, Lianli Gao, Hengtao Shen, and Jingkuan Song. Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. *arXiv preprint arXiv:2508.06426*, 2025.

Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. *arXiv preprint arXiv:2503.08558*, 2025.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Sergey Zakharov, Wadim Kehl, and Slobodan Ilic. Deceptionnet: Network-driven domain randomization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

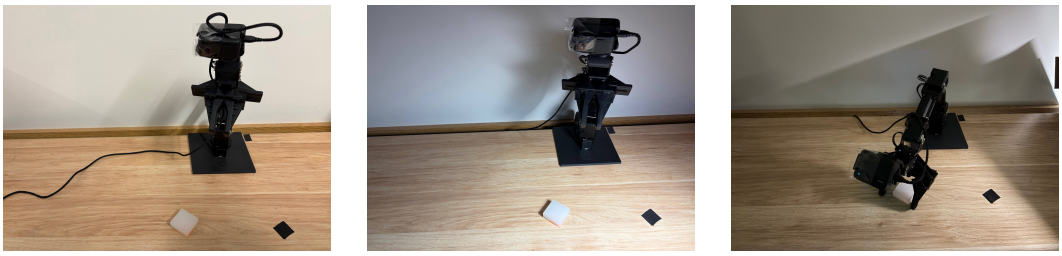
Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2304.13705.

A Appendix

A.1 Hardware Evaluation Details

We evaluate an ACT imitation learning policy (Zhao et al., 2023) on a hardware cube pick-and-place task under three lighting conditions (Fig. 7): *Base*, which matches the training distribution, and *Bright*

and *Dark*, which introduce out-of-distribution (OOD) illumination changes. We compare a baseline trained without Lightness DR against a policy trained with Lightness DR. Each condition is evaluated over 10 trials.



(a) Base (b) Bright (OOD) (c) Dark (OOD)

Figure 7: Real-world evaluation conditions. The Base setting matches training; Bright and Dark are out-of-distribution lighting variations.

Table 3 reports success rates. Without Lightness DR, performance collapses from 6/10 to 1/10 under both OOD conditions. With Lightness DR, the policy reaches 10/10 in-distribution and retains 8/10 and 6/10 under Bright and Dark lighting, respectively.

Table 3: Real-world success rates over 10 trials per condition. Parenthesized values give the relative drop from the Base condition. Lightness DR improves OOD success from 1/10 to 8/10 (Bright) and 6/10 (Dark), while also improving in-distribution success.

	Base	Bright (OOD)	Dark (OOD)
Without Lightness DR	6/10	1/10 (−83%)	1/10 (−83%)
With Lightness DR	**10/10**	**8/10** (−20%)	**6/10** (−40%)

A.2 eNTK vs. Activation Similarity as a Predictor of Transfer

A natural alternative to the empirical neural tangent kernel (eNTK) is to measure similarity between inputs directly in a network's activation space. Here we test which of the two better predicts how training on one input changes the model's output on another.

Setup. We use the square-size CNN of Sec. 3, taking one checkpoint per $N_{\text{color}} \in \{1, \dots, 27\}$. For each checkpoint, we take a single small SGD step on a probe input x and measure the induced change $\Delta f(x')$ in the output on held-out probes x', yielding approximately 1.7k (x, x') pairs per checkpoint. We then compute the Spearman rank correlation between $\Delta f(x')$ and each candidate similarity measure.

Similarity measures. The eNTK is

$$K(x, x') = \nabla_\theta f(x)^\top \nabla_\theta f(x'), \tag{2}$$

which, to first order in the step size η, governs the output change on x' after a gradient step on x:

$$\Delta f(x') \approx -\eta\, K(x', x)\, \nabla_f \mathcal{L}\big(f(x)\big). \tag{3}$$

For activation similarity, we compute the cosine similarity between the representations of x and x' at every layer of the network, reporting both the MLP head and the layer that achieves the highest correlation (chosen post hoc, which favors this baseline).

Results. Table 4 shows that the eNTK nearly determines the measured transfer ($\rho = 0.97 \pm 0.06$). Activation similarity is weak at every layer: even the best layer, selected with access to the outcome, reaches only $\rho = 0.22 \pm 0.28$, and the MLP head is close to uncorrelated with high variance across checkpoints ($\rho = 0.09 \pm 0.52$). Two inputs can therefore have similar representations while being

Table 4: Predicting training transfer from similarity. Spearman ρ between similarity $\mathrm{sim}(x, x')$ and measured output change $\Delta f(x')$ after one SGD step on x, reported as mean $\pm$ s.d. across 27 checkpoints ($\approx$1.7k pairs each). The best activation layer is chosen post hoc.

Similarity space	Spearman $\rho\left(\mathrm{sim}(x, x'),\ \Delta f(x')\right)$
eNTK $K(x, x')$	$\mathbf{+0.97 \pm 0.06}$
Activations (best layer)	$+0.22 \pm 0.28$
Activations (MLP head)	$+0.09 \pm 0.52$

coupled only weakly through shared parameters, and vice versa. Because transfer under gradient descent depends on how parameter gradients align rather than on how representations align, the eNTK is the appropriate object for predicting cross-input effects of training.