Modeling Imitation Learning Robustness to Noisy Demonstrations via Sigmoid Degradation

Everett Richards[1], Liu Dai[2]
[1]*San Diego State University,* [2]*University of California San Diego*
{ehrichards@sdsu.edu, l2dai@ucsd.edu}

***Abstract*—Behavioral Cloning (BC) is a simple and widely used approach in Imitation Learning (IL), but its performance is highly sensitive to the quality of demonstration data. In this work, we evaluate the robustness of BC policies trained on noisy demonstrations generated by MimicGen, a synthetic data generation system that augments a small number of human demonstrations through scene variation and trajectory stitching. We systematically inject Gaussian spatial noise into the action trajectories of MimicGen-generated data and train BC policies across 11 robotic manipulation tasks. We find that small amounts of noise improve generalization, whereas higher levels predictably degrade policy success. To characterize this behavior, we introduce a four-parameter sigmoid model that captures the relationship between noise amplitude and downstream policy performance. Our model achieves an aggregate R^2 of 0.9962, and can be estimated using just one or two data points with under 4% error. This framework offers a lightweight, quantitative tool for assessing demonstration quality and robustness in IL pipelines, supporting safer deployment of automated systems in domains where clean supervision data may be limited.**

***Index Terms*—Robotics, Imitation Learning, Gaussian Noise, Robustness, Sigmoid**

I. Introduction

Imitation Learning (IL) enables robots to learn complex behaviors by mimicking demonstrations rather than relying on hand-engineered reward functions or extensive trial-and-error. Among the most accessible forms of IL is Behavioral Cloning (BC), which uses supervised learning to train a policy that maps observations to actions. While simple and scalable, BC is known to be highly sensitive to the quality and distribution of its training data. In practical robotics settings, such as teleoperation, in-home assistance, or remote exploration, demonstrations may be noisy due to operator delay, sensor jitter, or labeling inconsistencies. Understanding how such noise affects downstream policy performance is crucial for robust deployment.

To address data scarcity in IL, recent systems like **MimicGen** [1] automate the generation of diverse demonstrations using a limited set of human-collected seeds. MimicGen works by segmenting demonstrations into object-centric subtasks, applying stochastic scene variation and geometric transformations, and stitching the segments into full trajectories. These generated demonstrations are then used to train BC policies. However, despite its scalability, MimicGen does not guarantee perfect trajectory fidelity, and prior work has not explored how noisy or perturbed demonstrations affect the resulting policy.

In this paper, we examine how Gaussian noise injected into MimicGen-generated demonstrations influences the performance of BC-trained policies across a variety of manipulation tasks. Specifically, we apply controlled spatial perturbations to the action sequences within MimicGen's synthetic demonstrations and use them to train BC policies from scratch. We then evaluate each trained policy's success rate on a fixed test set of unperturbed demonstrations.

Our experimental pipeline proceeds as follows:

1) Generate synthetic demonstrations using MimicGen from a small set of human seeds.
2) Inject isotropic Gaussian noise into each trajectory's 3D spatial components with varying standard deviations (σ).
3) Train separate BC policies on each noisy dataset.
4) Evaluate each policy and record its success rate.

To model the relationship between noise level and downstream performance, we fit a four-parameter sigmoid function that captures early-stage robustness, eventual degradation, and asymptotic performance bounds. We further show that this function can be estimated with high accuracy using only one or two observed performance points, enabling lightweight robustness diagnostics without exhaustive retraining.

This analysis is conducted across 11 manipulation tasks and 30 distinct noise levels. Our results show that while small amounts of noise can sometimes enhance generalization, performance degrades predictably past a task-dependent inflection point. These trends are consistent and well-captured by our proposed sigmoid model, with an aggregate R^2 of 0.9962 and prediction errors under 4% when using minimal data.

Contributions:

- A systematic study of how trajectory-level Gaussian noise in MimicGen demonstrations affects BC policy performance across 11 tasks.
- A four-parameter sigmoid model that accurately captures the noise-performance relationship.
- A predictive framework for estimating robustness with minimal data, offering a practical tool for evaluating demonstration quality in IL pipelines.

II. Related Works

A. Imitation Learning and Demonstration Quality

Imitation Learning (IL), and particularly Behavioral Cloning (BC), enables robots to learn visuomotor policies from demonstrations. While BC is simple and scalable, its performance is known to degrade under distribution shift or imperfect supervision [2]. Several methods have explored ways to augment or densify demonstration data, including Diffusion Policy [3], which models trajectories as conditional denoising diffusion processes. However, most IL studies either assume clean demonstrations or focus on large-scale data augmentation—leaving open the question of how noisy supervision affects downstream policy performance.

B. Noise Injection in Neural Networks

Injecting noise during training is a classic and effective regularization technique, shown to improve generalization and encourage exploration of flatter minima. Zhou *et al.* (2019) proved that stochastic perturbations can help escape local optima in training deep networks [4]. More recently, Bayesian approaches have framed noise injection as approximate inference, using techniques like Monte Carlo noise addition to estimate uncertainty alongside performance [5].

C. Robustness in Robotics

Real-world robot deployments frequently face scenarios with corrupted sensor input, inaccurate control, or mislabeled feedback. Prior research has examined how training under noise, especially label noise, can both undermine policy learning or, paradoxically, bolster robustness. For instance, models exposed to stochastic label perturbations early in training demonstrate increased resilience to later noise injection [6], [7]. While noise-resistant learning methods are prevalent in perception and classification domains, formal, quantitative modeling of noise impact in imitation learning, especially using synthetic pipelines like MimicGen, has not been previously performed. Our work fills this gap with a structured regression-based analysis.

D. Synthetic Demonstration Frameworks

Several recent systems have emerged to generate synthetic demonstration data from limited human inputs, enabling data-efficient robot training:

MimicGen uses a few human demonstrations, segments them into object-centric sub-tasks, transforms these segments to new scenes, and stitches them into full demonstrations. It has produced over 50,000 demonstrations across 18 tasks with just 200 human examples, and has been shown to match or exceed performance obtained with fully human-collected data [1].

DemoGen adapts single human demonstrations into many synthetic ones by spatially transforming trajectory segments and rendering visual inputs using 3D point-cloud editing. DemoGen has improved visuomotor generalization on real-world setups with only one source demo [8].

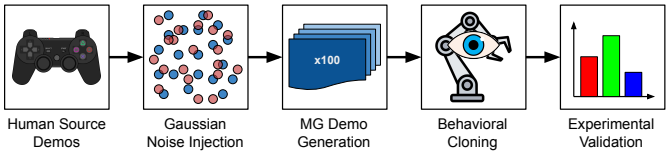

Fig. 1. Experimental pipeline for robustness evaluation. Gaussian noise is injected into source demonstrations before MimicGen augmentation. BC policies are trained on the resulting datasets and evaluated on clean tasks to measure noise-induced performance degradation.

SkillMimicGen builds on this concept by segmenting demonstrations into reusable skills and recombining them across contexts using motion planning and a hybrid skill policy. It produced over 24,000 skill-based demonstrations and demonstrated zero-shot sim-to-real transfer, outperforming MimicGen by roughly 24% in success rates [9].

E. Our Contribution

Unlike prior work that introduces new demonstration-generation systems or treats noise primarily as a tool for data augmentation, our contribution lies in rigorously analyzing how Gaussian spatial noise impacts task performance in existing imitation learning pipelines like MimicGen. Rather than proposing a new generator, we focus on modeling robustness: we introduce a parameterized sigmoid function that quantitatively captures how task success rates evolve with increasing noise. This model enables interpretable, task-level assessments of noise sensitivity and allows performance to be estimated from just one or two data points, making it lightweight and practical for real-world diagnostics. Our work fills a critical gap by positioning noise not as a nuisance, but as a measurable and sometimes beneficial factor, offering insights at the intersection of data augmentation, uncertainty modeling, and scalable robot learning.

III. Method

A. Problem Definition

Consider a robotic actuator designed to perform task $\mathcal{M}$. Suppose we have source demonstrations $\tau^{(j)} \in \mathcal{D}_{\mathcal{M},src}$. Suppose we inject Gaussian noise with standard deviation σ into each 3D position component of each source demonstration. Our goal is to model the relationship between the level of Gaussian noise σ injected into synthetic demonstrations and the resulting performance of a BC policy trained on those demonstrations. For each task $\mathcal{M}$, we inject Gaussian noise into the spatial components of MimicGen-generated trajectories and use the resulting dataset to train a BC policy from scratch. We then evaluate the trained policy's success rate on a fixed set of clean validation scenes. We model this relationship using a parameterized sigmoid function $S_{\mathcal{M}}(\sigma)$ that maps noise amplitude to task success rate $\rho_{\mathcal{M}}$.

We model the relationship between noise magnitude σ and performance ρ according to Eq. 1:

$$S(\sigma) = \lambda \left(\frac{1}{\delta + e^{\kappa\sigma + \chi}} \right) \tag{1}$$

Where λ, δ, κ, and χ represent the scaling factor, vertical offset, steepness, and horizontal shift, respectively. This 4-parameter sigmoid captures initial improvement, performance plateau, and degradation. We fit this using full-curve optimization and a 1-2 point linear approximation. This allows us to determine the distribution's key aspects without having to perform extensive experiments.

B. One-Dimensional (1D) Sigmoid Parameter Estimation

In the case that no noise experiments are conducted, we can still develop a sigmoid model with parameters approximated by using the model's baseline performance. Let ρ_0 be the performance of a model when no noise is injected, i.e. $\rho_0 = \rho(0)$. This will serve as an approximate baseline difficulty for the task $\mathcal{M}$. Since ρ_0 is typically among the highest values in the dataset, it is reasonable to set the scale $\lambda := \rho_0$, whose value is given. Then, we will optimize parameters δ, κ, and χ according to the following linear system:

$$\begin{bmatrix}\lambda\\ \delta\\ \kappa\\ \chi\end{bmatrix} = \begin{bmatrix}0 & 1\\ \delta_0 & \delta_1\\ \kappa_0 & \kappa_1\\ \chi_0 & \chi_1\end{bmatrix}\begin{bmatrix}1\\ \rho_0\end{bmatrix} = A_{1D}\vec{x}_{1D} \tag{2}$$

C. Two-Dimensional (2D) Sigmoid Parameter Estimation

In the case that only limited noise experiments are performed, we can formulate a better approximation for the sigmoid function's parameters by having some insight into the model's behavior. Let ρ_1 be the performance of a model when a moderate amount of noise is injected. We assume that this moderate noise threshold is given, although it varies depending on the task. We optimize the model's parameters according to the following linear system:

$$\begin{bmatrix}\lambda\\ \delta\\ \kappa\\ \chi\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ \delta_0 & \delta_1 & \delta_2\\ \kappa_0 & \kappa_1 & \kappa_2\\ \chi_0 & \chi_1 & \chi_2\end{bmatrix}\begin{bmatrix}1\\ \rho_0\\ \rho_1\end{bmatrix} = A_{2D}\vec{x}_{2D} \tag{3}$$

D. Noise Injection Procedure

We inject Gaussian noise into the action trajectories of source demonstrations, with standard deviation σ, as seen in Eq. 4 and Alg. 1.

$$\epsilon \sim \mathcal{N}(0, \sigma^2), \quad \tau_i \leftarrow \tau_i + \epsilon \tag{4}$$

After injecting noise, we use the resulting demonstrations to train a behavioral cloning (BC) policy using supervised learning, as illustrated in Fig. 1. Each policy is trained independently for a given noise level and task.

E. Experimental Setup and Policy Training

We evaluate 11 robotic manipulation tasks using demonstrations generated by MimicGen. For each task, we begin with 10 human-provided seed demonstrations, from which MimicGen generates 100 synthetic demonstrations by segmenting trajectories, applying scene variation, and stitching object-centric subtasks.

Algorithm 1: Noise Injection Experimental Procedure

Data: Task $\mathcal{M}$, noise level σ, trials n, dataset $\mathcal{D}_{src}$
Result: Success rate R
Function *NoiseExperiment*($\mathcal{M}, \sigma, n, \mathcal{D}_{src}$)**:**
  $\mathcal{D}_{noisy} \leftarrow \emptyset$
  foreach *trajectory* $\tau \in \mathcal{D}_{src}$ **do**
    Inject noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to τ
    Append to $\mathcal{D}_{noisy}$
  end
  Generate demos with MimicGen
  Count successes and return success rate R
end

To simulate imperfect demonstrations, we inject Gaussian spatial noise with standard deviations in the range $[0.00, 0.50]$ into the 3D action trajectories of the generated demonstrations. For each noise level σ, we train a behavioral cloning (BC) policy from scratch using the corresponding noisy dataset. Policies are implemented using a feedforward neural network matching the original MimicGen architecture and are trained using standard supervised learning.

Each trained policy is evaluated on a separate validation set of 100 clean demonstrations. We define the **success rate** as the proportion of validation episodes in which all task goals are completed successfully.

After collecting success rates across all noise levels for each task, we fit a sigmoid degradation curve using the `curve_fit` function from `scipy.optimize`. We extract the fitted parameters $(\lambda, \delta, \kappa, \chi)$ and then solve for matrices A_{1D} and A_{2D} using `LinearRegression` from `sklearn.linear_model`, as defined in Eq. 2 and Eq. 3.

IV. Experimental Results

A. Sigmoid Matrix Parameters

After optimizing the parameters through a linear equation according to the sigmoid function $S_{\mathcal{M}}$, based on zero-noise performance $\rho_0 = \rho(0.00)$ across all 11 tasks, we obtained the following parameters:

$$\begin{bmatrix}\lambda\\ \delta\\ \kappa\\ \chi\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1.0387 & -0.0010\\ 17.6122 & -0.0162\\ -3.1988 & -0.0015\end{bmatrix}\begin{bmatrix}1\\ \rho_0\end{bmatrix}$$

When using two points $\rho_0 = \rho(0.00)$ and $\rho_1 = \rho(0.25)$ to fit the Sigmoid curve, we obtained:

$$\begin{bmatrix}\lambda\\ \delta\\ \kappa\\ \chi\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0.9580 & 0.0004 & 0.0026\\ 15.1902 & 0.1352 & -0.1751\\ -2.6760 & -0.0107 & -0.0238\end{bmatrix}\begin{bmatrix}1\\ \rho_0\\ \rho_1\end{bmatrix}$$

As expected, these matrices indicate that the accuracy-noise distribution can be appropriately modeled by a negated sigmoid function with a horizontal offset.

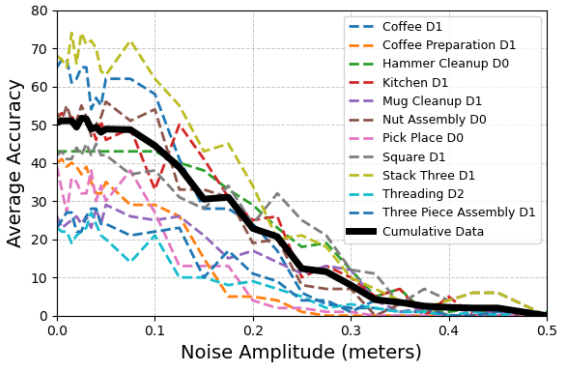

Fig. 2. Success rate ρ vs. noise level σ for 11 tasks. Cumulative average shown as bold black line.

B. Computed Sigmoid Functions

The relationship between noise amplitude and average accuracy across 11 tasks is illustrated in Fig. 2. Computed sigmoid functions, including fitted sigmoids and 1D/2D predicted sigmoids, are visualized in Fig. 3. Accuracy refers to the success rate of a behavioral cloning policy trained on noisy MimicGen demonstrations and evaluated on clean data.

C. Model Fitness Analysis

To evaluate model fitness, we report four standard regression metrics in Tables I, II, and III: mean absolute error (MAE), sum of squared errors (SSE), root mean squared error (RMSE), and the coefficient of determination (R^2). MAE and RMSE quantify prediction error in intuitive units, with RMSE placing greater emphasis on larger deviations, while SSE reflects total squared deviation. R^2 measures how well the sigmoid model explains variance in the data. While all four are included for completeness, we primarily emphasize R^2 and RMSE as they best capture trend fidelity and robustness to error. The "Cumulative" row in each table reflects the arithmetic mean of success rates across all 11 tasks at each noise level, with a sigmoid fit applied to this aggregate curve to represent average-case behavior.

Task	MAE	SSE	RMSE	R^2
Square	2.37	268	3.04	0.9630
Nut Assembly	2.30	280	3.11	0.9814
Three Piece Assembly	1.70	145	2.24	0.9563
Kitchen	2.37	362	3.53	0.9720
Coffee	2.60	354	3.50	0.9841
Stack Three	2.87	337	3.41	0.9856
Hammer Cleanup	**0.99**	**51**	1.33	0.9943
Pick Place	1.97	264	3.02	0.9626
Coffee Preparation	1.37	114	1.98	0.9867
Mug Cleanup	1.57	123	2.06	0.9601
Threading	1.46	129	2.11	0.9501
Cumulative	0.969	48.1	**1.29**	**0.9962**

TABLE I
SUMMARY STATISTICS FOR FITTED SIGMOID

Task	MAE	SSE	RMSE	R^2
Square	5.30	1542	7.29	0.7870
Nut Assembly	2.44	343	3.44	0.9773
Three Piece Assembly	2.65	334	3.39	0.8993
Kitchen	3.41	628	4.65	0.9514
Coffee	3.45	591	4.52	0.9735
Stack Three	4.18	826	5.34	0.9648
Hammer Cleanup	4.68	1121	6.22	0.8760
Pick Place	4.30	972	5.79	0.8662
Coffee Preparation	3.48	725	5.00	0.9150
Mug Cleanup	3.00	499	4.15	0.8374
Threading	**1.55**	**166**	**2.39**	0.9355
Cumulative	2.39	222	2.77	**0.9823**

TABLE II
SUMMARY STATISTICS FOR 1D SIGMOID

Task	MAE	SSE	RMSE	R^2
Square	3.16	470	4.03	0.9350
Nut Assembly	2.93	491	4.12	0.9674
Three Piece Assembly	2.85	385	3.64	0.8840
Kitchen	3.80	784	5.20	0.9393
Coffee	2.84	428	3.84	0.9800
Stack Three	3.48	629	4.66	0.9732
Hammer Cleanup	2.58	263	3.01	0.9709
Pick Place	2.89	490	4.11	0.9308
Coffee Preparation	1.87	197	2.61	0.9769
Mug Cleanup	2.36	312	3.28	0.8984
Threading	**1.59**	**155**	**2.31**	0.9396
Cumulative	2.14	174	2.45	**0.9861**

TABLE III
SUMMARY STATISTICS FOR 2D SIGMOID

V. DISCUSSION

Our results reveal several consistent trends across the 11 evaluated tasks. Injecting low-magnitude Gaussian noise ($\sigma < 0.05$) into MimicGen-generated demonstrations often improves the downstream performance of behavioral cloning (BC) policies, likely due to regularization effects that encourage generalization and reduce overfitting. This aligns with established principles of data augmentation, where small perturbations enhance model robustness. However, as noise increases beyond a task-dependent threshold ($\sigma > 0.10$), performance declines sharply. This degradation can be attributed to distributional shift in the training data, where excessive noise introduces harmful bias and misleads the policy. These trends are accurately captured by our fitted sigmoid curves, which exhibit strong alignment with empirical performance data. Furthermore, our biparametric approximation—estimating the sigmoid from only two observed success rates—achieves under 4% average error, demonstrating the utility of our framework for low-cost robustness estimation.

The sigmoid model introduced in this work provides a compact and interpretable summary of each task's robustness profile. The scale parameter λ reflects task difficulty under clean data; δ captures the degradation floor; and the slope and shift terms (κ, χ) determine sensitivity to noise. Together, these parameters quantify the extent and onset of performance collapse and enable lightweight performance prediction.

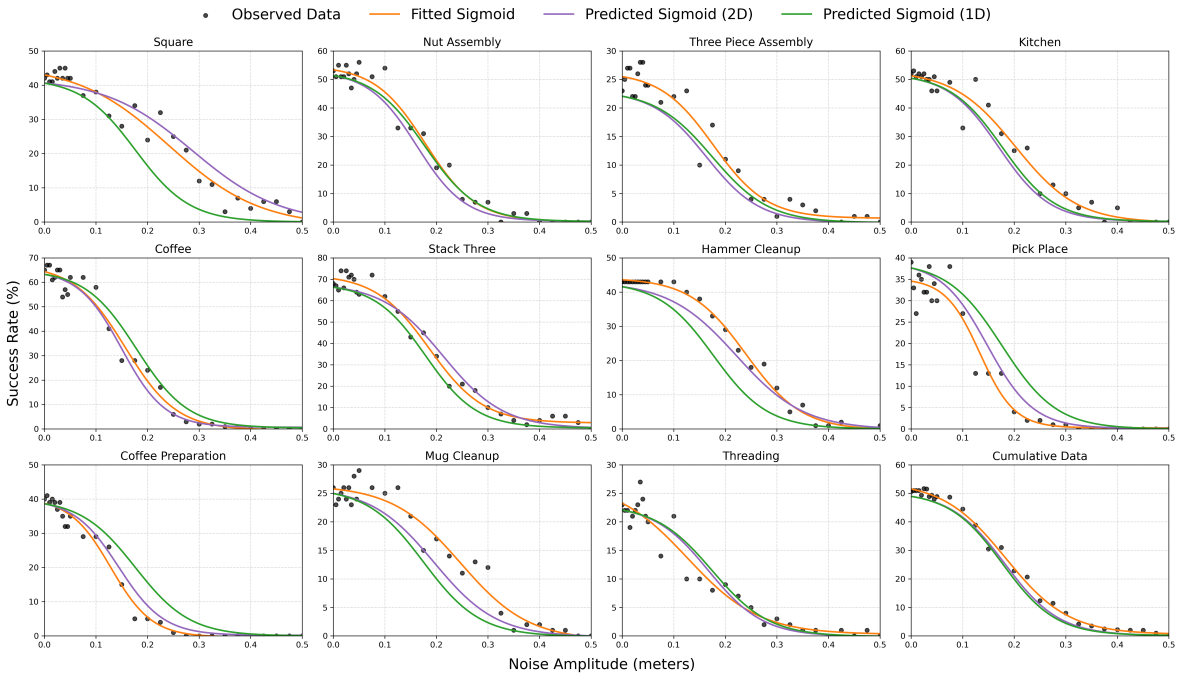

Fig. 3. Parameterized sigmoid models to approximate noise-accuracy tradeoff, across 11 robot manipulation tasks and cumulative data. Best-fit sigmoids are shown in black, while the sigmoids predicted by univariate and bivariate models are shown in green and purple, respectively.

This study is limited to Gaussian spatial noise applied during training on synthetic demonstrations in simulation. Future work should extend the framework to other noise types (e.g., sensor, temporal) and validate its applicability to real-world hardware and other IL pipelines beyond MimicGen.

VI. Conclusion

This work presents a quantitative study of how noise in synthetic demonstrations affects the downstream performance of behavioral cloning in imitation learning. By fitting a four-parameter sigmoid to success-rate data collected across 11 manipulation tasks and 30 noise levels, we found that small amounts of noise can improve generalization, but excessive noise predictably degrades performance. Crucially, our sigmoid model generalizes well. The fitted version achieves an aggregate R^2=0.9962, while our one- and two-point predictive variants maintain less than 4% average error, demonstrating that accurate robustness diagnostics can be obtained with minimal experimentation. This approach enables practitioners to assess the noise tolerance of their pipelines quickly, guiding decisions about acceptable data quality and whether controlled noise injection could be beneficial. Future work could extend this methodology to other noise modalities (e.g., sensor or policy noise) and validate findings in real-world robotic platforms. As synthetic demonstration pipelines like MimicGen expand into sim-to-real and human-in-the-loop systems, principled noise modeling lays a foundation for robust, resource-efficient robot learning.

Acknowledgment

This work was supported in part by NSF Grant #2150643 through the REU Site in Interdisciplinary Artificial Intelligence at the University of California San Diego.

References

[1] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," 2023. [Online]. Available: https://arxiv.org/abs/2310.17596

[2] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," 2011. [Online]. Available: https://arxiv.org/abs/1011.0686

[3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," 2024. [Online]. Available: https://arxiv.org/abs/2303.04137

[4] M. Zhou, T. Liu, Y. Li, D. Lin, E. Zhou, and T. Zhao, "Toward understanding the importance of noise in training neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 7594–7602. [Online]. Available: https://proceedings.mlr.press/v97/zhou19d.html

[5] X. Yuan, J. Li, and E. E. Kuruoglu, "Uncertainty quantification with noise injection in neural networks: A bayesian perspective," 2025. [Online]. Available: https://arxiv.org/abs/2501.12314

[6] H. Wei, L. Tao, R. Xie, and B. An, "Open-set label noise can improve robustness against inherent label noise," 2021. [Online]. Available: https://arxiv.org/abs/2106.10891

[7] P. Chen, G. Chen, J. Ye, J. Zhao, and P.-A. Heng, "Noise against noise: stochastic label noise helps combat inherent label noise," in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://iclr.cc/virtual/2021/spotlight/3537

[8] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu, "Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning," 2025. [Online]. Available: https://arxiv.org/abs/2502.16932

[9] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," 2024. [Online]. Available: https://arxiv.org/abs/2410.18907