Ambient Diffusion Policy: Imitation Learning from Suboptimal Data in Robotics

Adam Wei, Nicholas Pfaff, Thomas Cohn, Arif Kerem Dayı,
Constantinos Daskalakis, Giannis Daras, Russ Tedrake
MIT
https://ambient-diffusion-policy.github.io/

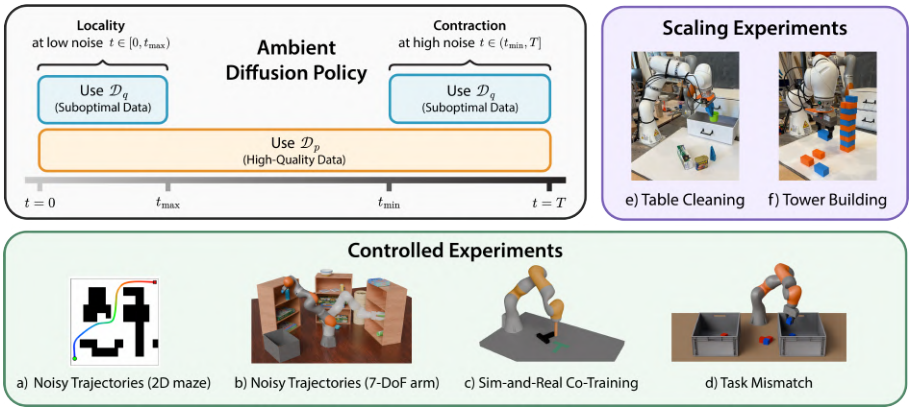


Fig. 1: **Training Algorithm for Ambient Diffusion Policy.** High-quality data can be used during training at all diffusion times. Suboptimal data is restricted to training at low or high diffusion times $t \in [0, t_{\max}) \cup (t_{\min}, T]$. **Task visualizations.** We evaluate Ambient Diffusion Policy on four types of action suboptimality, six different tasks, and scale to the Open X-Embodiment (OXE) [1].

I. Introduction

The training corpus of nearly every large-scale robot policy spans different tasks, real and simulated environments, embodiments, and modalities [2, 3, 4, 5, 1, 6, 7, 8, 9]. High-quality, task-specific data is expensive to collect, while suboptimal data is abundant: real-world collection naturally produces failures and trajectories of differing quality [10, 1, 8, 11], and out-of-distribution sources (simulation, cross-embodied data, egocentric video) are plentiful [12, 13, 1, 14, 15]. Practitioners routinely combine sources of varying quality into massive pretraining sets [1, 8, 5, 16, 11], yet learning from arbitrary suboptimal or shifted distributions remains underexplored in robotics [5, 2, 12, 16].

Discarding the lowest-quality samples is wasteful since even suboptimal samples contain useful learning signal [19]. The most common alternative is to "co-train" on everything and re-weight the contribution of each dataset [2, 3, 4, 5, 12, 13]; however, this trains the policy to sample from a mixture with low-quality distributions and thus fundamentally biases the policy [20, 10]. Finetuning is complementary but can be insufficient alone (Appendix O).

Our key intuition is that suboptimal and high-quality samples differ in some features but align in others. For example, non-expert teleoperators may share the same high-level plan as experts but exhibit less precise manipulation; conversely, pick-and-place data may be useful for grasping primitives while encoding the wrong task. We turn this intuition into an algorithm by observing that robot action data exhibits a spectral power law [21], which induces a *global-to-local hierarchy* in action diffusion and a *locality* property [22] in the optimal denoisers. Concretely, Diffusion Policy [23] learns high-level planning at high noise and motion primitives at low noise. A policy can therefore selectively absorb useful features from suboptimal data by using it only at noise levels where it aligns with the target distribution. This reveals a new design axis for co-training in robotics: *noise-dependent data usage*.

To this end, we propose **Ambient Diffusion Policy**, a principled algorithm for training Diffusion Policies on suboptimal robot data, in which each suboptimal sample contributes only at the high or low diffusion times where it aligns with the target. Our method extends the Ambient Diffusion Omni framework [24, 19] ("Ambient" for short), previously applied to vision and protein design [25]. Our contributions are:

- **Properties of Robot Data.** We show empirically that robot action data exhibits a spectral power law, inducing a *global-to-local hierarchy* and *locality* in action diffusion that make Ambient well-suited for robotics (Appendix A).
- **Ambient Diffusion Policy.** A simple, principled method for learning from suboptimal data that requires just a *single change* to Diffusion Policy's data sampler (Section IV).

- **Generality.** Ambient outperforms baselines on three common types of action suboptimality: noisy demonstrations, sim-to-real gap, and task mismatch (Appendix J).
- **Scale.** On Open X-Embodiment [1]—a large dataset with mixed quality and unstructured distribution shifts—Ambient beats co-training by up to **33%** on two real-world tasks, and keeps improving as the suboptimal data scales while co-training plateaus (Section V).
- **Theory.** For a simple model, we prove the spectral power law implies fast contraction through noise and locality of the optimal policies, justifying our algorithm and advancing the theory of the broader Ambient framework (Appendix B).

II. Related Work

Re-weighting and Filtering. The prevailing approach to heterogeneous robot datasets is to re-weight each dataset's contribution to the training loss [2, 3, 4, 5, 6, 7, 12, 13, 26], i.e. minimizing $\mathcal{L}_{\text{co-training}} = \alpha\,\mathcal{L}_{\mathcal{D}_p} + (1-\alpha)\,\mathcal{L}_{\mathcal{D}_q}$ for a weight $\alpha \in [0,1]$; we refer to this standard approach as "co-training." Beyond the difficulty of selecting weights [7, 5, 28, 29, 16, 30], even with the optimal weights, co-training biases the policy, since it is still trained to sample from a mixture containing suboptimal distributions [20, 10]. A common alternative is to filter suboptimal samples entirely [5, 2, 17, 31, 32], but this is wasteful: even low-quality samples carry useful signal [19, 11, 10]. For example, Dexora [17] discards 80% of its demonstrations and OpenVLA's Magic Soup++ [5] uses only 27 of the datasets in OXE [1]; our method instead extracts useful signal from the discarded data (Appendix K, Section V). It avoids both issues by training on suboptimal samples only within specific intervals of the diffusion process. Importantly, these intervals have precise interpretations and can be automatically annotated.

Learning from Suboptimal or OOD Data. Prior works on learning from corrupted data [33, 24, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44] often require multiple training runs [45, 46, 47] or knowledge of the corruption, which is unrealistic in robotics (e.g. the sim-to-real gap has no closed-form description). In robotics, $\pi_{0.7}$ [10] labels data quality and LDA-1B [48] relegates low-quality trajectories to an auxiliary objective; others learn invariant representations [49, 50] but are hard to scale [12]. Robust imitation learning [20, 51] *ignores* corrupted samples rather than learning from them, and needs over half the data to be high-quality. Concurrent work explores a similar idea for hand-tracking data [52]. In contrast, our method learns from arbitrary mixtures of suboptimal action data, scales to large real-world datasets [1], has theoretical grounding, and is simple to implement.

III. Background

A. Robot Imitation Learning and Diffusion Policy

Given a dataset $\mathcal{D}_p = \{(O^{(i)}, A^{(i)})\}_{i=1}^{n_p}$ of observation-action pairs, imitation learning trains a policy $\pi(A \mid O)$ that approximates the conditional $p(A \mid O)$ [23], where A and O are chunks of future actions and recent observations [53]. Diffusion Policy [23] samples from $\pi(A \mid O)$ with a conditional denoising process over the action space [54]. It learns a family of denoisers $\{h_\theta(A_t, O, t)\}_{t\in[0,T]}$ that predict the clean action A_0 from a noisy version $A_t = A_0 + \sigma_t Z$ with $Z \sim \mathcal{N}(0, I)$. Minimizing the denoising loss [54] over $t \sim \mathcal{U}[0,T]$ and $(O, A_0) \sim \mathcal{D}_p$ yields the conditional expectation $h^*_\theta(A_t, O, t) = \mathbb{E}[A_0 \mid A_t, O]$, from which one samples by running the reverse process [23, 57].

B. Problem Setting

Suppose we have a (small) dataset $\mathcal{D}_p$ from a target distribution p and a much larger dataset $\mathcal{D}_q$ from a shifted or suboptimal distribution q. As discussed above, this is a common setting in robotics.

Problem Statement: Our goal is to leverage the vast pool of *suboptimal* samples in $\mathcal{D}_q$ to improve the estimation of the denoisers, $\mathbb{E}_p[A_0 \mid A_t, O]$, on the user-defined *target* distribution.

The definitions of p and q are arbitrary; here $\mathcal{D}_p$ is expert demonstrations on the target robot, environment, and task, though any definition applies (e.g. heuristic measures [58] or human labels [10]). We construct $\mathcal{D}_q$ from controlled distribution shifts (Appendix J) and from Open X-Embodiment (Section V) [1].

C. Ambient Diffusion Omni

This Problem Statement is not unique to robotics. Ambient Diffusion Omni [19] addresses a similar problem in vision: it allows suboptimal samples from q to be used only at certain diffusion times. The algorithm assigns two parameters, $t_{\min}$ and $t_{\max}$, to samples from $\mathcal{D}_q$, which may then contribute to learning only at diffusion times $t \in [0, t_{\max}) \cup (t_{\min}, T]$.

The key mechanism is *noise as contraction*: for any p and q supported on a subset of $\mathbb{R}^d$ with diameter D, $\mathrm{d_{TV}}(p_t, q_t) \le \mathrm{d_{TV}}(p, q) \cdot D/(2\sigma_t)$, where $p_t = p \circledast \mathcal{N}(0, \sigma_t I)$ and $q_t = q \circledast \mathcal{N}(0, \sigma_t I)$. Intuitively, noise erases distributional differences: p and q *contract* toward one another, so there is a sufficiently high time $t_{\min}$ above which noisy actions from the two distributions are effectively indistinguishable and $\mathcal{D}_q$ can safely contribute. The utility of q is highest when $t_{\min}$ is small (fast contraction).

Section IV examines structural properties of robot action data that make this method well-suited for robotics, then presents our algorithm. Appendix A expands on these properties and Appendix B theoretically formalizes them, building on prior work [19, 25]. Despite the in-depth motivation, the method is remarkably simple: it requires a **single change** to the Diffusion Policy [23] data sampler.

IV. Ambient Diffusion Policy: Method

Ambient Diffusion is well suited for robotics due to the structure of robot data. Most importantly, robot data exhibits

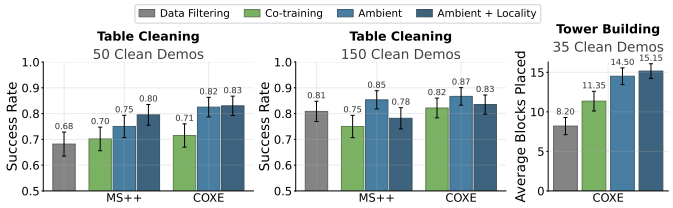


Fig. 2: **When scaled to OXE, Ambient Diffusion Policy outperforms data filtering and co-training** by up to 15% on table cleaning and 84% on tower building (20 trials per policy).

a *spectral power law* (Appendix A). This induces two properties that our algorithm exploits. The first is a *global-to-local hierarchy*: Diffusion Policies learn high-level planning at high noise and low-level motion primitives at low noise. The second is *locality* of the optimal denoisers: at low noise, each predicted action depends mainly on its temporal neighbors. Together they justify *noise-dependent data usage*: suboptimal data is safe at high noise, where noise contracts p and q (Section III) and masks local suboptimality, and at low noise, where locality lets the policy copy shared primitives without inheriting incorrect global structure. We discuss details further in the Appendix.

We now present our algorithm: Ambient Diffusion Policy. It has two phases: a) dataset annotation, and b) training.

A. Phase 1: Dataset annotation

This phase annotates $t_{\min}$ (and optionally $t_{\max}$) for the suboptimal samples from $\mathcal{D}_q$. The simplest method is a hyperparameter sweep. A more principled solution trains a classifier $c_\phi(A_t, t)$ that predicts whether a noisy action A_t came from p_t or q_t: $t_{\min}$ is the smallest noise level at which the classifier can no longer distinguish q_t from p_t (within a tolerance τ) [19]. If the classifier is well-trained, this guarantees distributional closeness between p_t and q_t for $t > t_{\min}$ (Appendix Theorem 4). The classifier can annotate at different granularities (per dataset vs. per sample; Appendix R), and a similar approach can annotate $t_{\max}$.

B. Phase 2: Training

Training is identical to Diffusion Policy [23] with one change: a custom data sampler. The sampler first draws a batch of diffusion times $t^{(i)} \sim \mathcal{U}[0, T]$, then draws admissible samples from $\mathcal{D}_p \cup \mathcal{D}_q$. Samples from $\mathcal{D}_p$ are always admissible; suboptimal samples from $\mathcal{D}_q$ are admissible only if $t^{(i)} \in [0, t_{\max}) \cup (t_{\min}, T]$. Figure 1 visualizes these intervals. Inference is unchanged. Full pseudocode, an optional alternative loss [24], and a training-time rejection-sampling variant are in Appendices H, G0a, and J.

V. Scaling Experiments: Open X-Embodiment

We show the generality of our approach across three controlled distribution shifts in Appendix J. In the main body, we choose to focus on our scaling experiments on Open X-Embodiment (OXE) [1], a large collection of real-world data with heterogeneous quality and unstructured distribution shifts. The goal is to extract useful signal from as much of this "suboptimal" data as possible, evaluated on two real-world tasks.

1) Table Cleaning (Figure 1e): the robot opens a drawer, places diverse (unseen) objects inside, and closes it; scored by the rubric in Appendix Table XI. **2) Tower Building** (Figure 1f): the robot stacks blocks in alternating directions and colors as high as possible, scored by the number of blocks placed before toppling.

The training procedure is identical for both tasks. $\mathcal{D}_p$ is a small number of task demonstrations; $\mathcal{D}_q$ is one of two OXE subsets: **Magic Soup++ (MS++)**, the 27 datasets used by OpenVLA [5], or **Custom OXE (COXE)**, a deliberately inclusive 48-dataset superset of MS++. We annotate $t_{\min}$ and $t_{\max}$ via Phase 1 (Appendix T).

Both methods re-weight each dataset's per-timestep sampling probability; for Ambient a dataset is only sampled at times within its $[0, t_{\max}) \cup (t_{\min}, T]$ interval (Appendix T).

A. Main Results

Table Cleaning ($\mathcal{D}_p$ with 50 demos): Filtering reaches 68.2% task completion; co-training adds only 2–3% and plateaus as $\mathcal{D}_q$ scales from MS++ to COXE. The best Ambient policy beats co-training by 12% and *improves* with the larger COXE; locality ($t_{\max} > 0$) helps further.

Table Cleaning ($\mathcal{D}_p$ with 150 demos): With more target data, filtering improves to 80.1% and suboptimal data adds less value, yet Ambient still beats co-training by up to 10% (locality no longer helps, as the policy already learns the primitives from $\mathcal{D}_p$).

Tower Building ($\mathcal{D}_p$ with 35 demos): Ambient policies build up to 84% and 33% taller towers than the filtering and co-training baselines, respectively (Figure 2); including $\mathcal{D}_q$ at low diffusion times helps.

B. Ablations

Three ablations are detailed in Appendix U. Re-weighting is optional for Ambient (a 3–9% gain) but *necessary* for co-training, whose unweighted policies were too unsafe to run. Finetuning on $\mathcal{D}_p$ gave no significant change (Appendix O), and Ambient policies need 25–40% fewer grasps and less time per object than co-training.

VI. Conclusion

High-quality, task-specific robot data is scarce, making it essential to learn from suboptimal data—yet filtering is wasteful, and co-training absorbs both the useful and the harmful parts of suboptimal samples. Building on the power of noise-dependent data usage in neighboring fields [19, 25], our key insight is that this paradigm is well-suited for robotics. Ambient Diffusion Policy makes no assumptions about the action suboptimality, outperforms existing baselines, and greatly expands the set of useful data sources for robot imitation learning. The question shifts from *which* data sources should be used, to *when* each should be used in the diffusion process.

REFERENCES

[1] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, M. Z. Irshad, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, V. Guizilini, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[2] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[3] NVIDIA, :, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[4] T. L. Team, J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, N. Kuppuswamy, K.-H. Lee, K. Liu, D. McConachie, I. McMahon, H. Nishimura, C. Phillips-Grafflin, C. Richter, P. Shah, K. Srinivasan, B. Wulfe, C. Xu, M. Zhang, A. Alspach, M. Angeles, K. Arora, V. C. Guizilini, A. Castro, D. Chen, T.-S. Chu, S. Creasey, S. Curtis, R. Denitto, E. Dixon, E. Dusel, M. Ferreira, A. Goncalves, G. Gould, D. Guoy, S. Gupta, X. Han, K. Hatch, B. Hathaway, A. Henry, H. Hochsztein, P. Horgan, S. Iwase, D. Jackson, S. Karamcheti, S. Keh, J. Masterjohn, J. Mercat, P. Miller, P. Mitiguy, T. Nguyen, J. Nimmer, Y. Noguchi, R. Ong, A. Onol, O. Pfannenstiehl, R. Poyner, L. P. M. Rocha, G. Richardson, C. Rodriguez, D. Seale, M. Sherman, M. Smith-Jones, D. Tago, P. Tokmakov, M. Tran, B. V. Hoorick, I. Vasiljevic, S. Zakharov, M. Zolotas, R. Ambrus, K. Fetzer-Borelli, B. Burchfiel, H. Kress-Gazit, S. Feng, S. Ford, and R. Tedrake. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL https://arxiv.org/abs/2507.05331.

[5] K. Kim, Moo Jin andwo, Z. Pan, et al. Openvla: An open-source vision-language-action model. In *arXiv preprint arXiv:2406.09246*, 2024.

[6] F. Lin, K. Arora, J. Mercat, H. Nishimura, P. Shah, C. Xu, M. Zhang, M. Zolotas, M. Angeles, O. Pfannenstiehl, A. Beaulieu, and J. Barreiros. A systematic study of data

modalities and strategies for co-training large behavior models for robot manipulation, 2026. URL https://arxiv.org/abs/2602.01067.

[7] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, S. Bohez, K. Bousmalis, A. Brohan, T. Buschmann, A. Byravan, S. Cabi, K. Caluwaerts, F. Casarini, O. Chang, J. E. Chen, X. Chen, H.-T. L. Chiang, K. Choromanski, D. D'Ambrosio, S. Dasari, T. Davchev, C. Devin, N. D. Palo, T. Ding, A. Dostmohamed, D. Driess, Y. Du, D. Dwibedi, M. Elabd, C. Fantacci, C. Fong, E. Frey, C. Fu, M. Giustina, K. Gopalakrishnan, L. Graesser, L. Hasenclever, N. Heess, B. Hernaez, A. Herzog, R. A. Hofer, J. Humplik, A. Iscen, M. G. Jacob, D. Jain, R. Julian, D. Kalashnikov, M. E. Karagozler, S. Karp, C. Kew, J. Kirkland, S. Kirmani, Y. Kuang, T. Lampe, A. Laurens, I. Leal, A. X. Lee, T.-W. E. Lee, J. Liang, Y. Lin, S. Maddineni, A. Majumdar, A. H. Michaely, R. Moreno, M. Neunert, F. Nori, C. Parada, E. Parisotto, P. Pastor, A. Pooley, K. Rao, K. Reymann, D. Sadigh, S. Saliceti, P. Sanketi, P. Sermanet, D. Shah, M. Sharma, K. Shea, C. Shu, V. Sindhwani, S. Singh, R. Soricut, J. T. Springenberg, R. Sterneck, R. Surdulescu, J. Tan, J. Tompson, V. Vanhoucke, J. Varley, G. Vesom, G. Vezzani, O. Vinyals, A. Wahid, S. Welker, P. Wohlhart, F. Xia, T. Xiao, A. Xie, J. Xie, P. Xu, S. Xu, Y. Xu, Z. Xu, Y. Yang, R. Yao, S. Yaroshenko, W. Yu, W. Yuan, J. Zhang, T. Zhang, A. Zhou, and Y. Zhou. Gemini robotics: Bringing ai into the physical world, 2025. URL https://arxiv.org/abs/2503.20020.

[8] AgiBot-World-Contributors, Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, S. Jiang, Y. Jiang, C. Jing, H. Li, J. Li, C. Liu, Y. Liu, Y. Lu, J. Luo, P. Luo, Y. Mu, Y. Niu, Y. Pan, J. Pang, Y. Qiao, G. Ren, C. Ruan, J. Shan, Y. Shen, C. Shi, M. Shi, M. Shi, C. Sima, J. Song, H. Wang, W. Wang, D. Wei, C. Xie, G. Xu, J. Yan, C. Yang, L. Yang, S. Yang, M. Yao, J. Zeng, C. Zhang, Q. Zhang, B. Zhao, C. Zhao, J. Zhao, and J. Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025. URL https://arxiv.org/abs/2503.06669.

[9] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[10] Physical Intelligence, B. Ai, A. Amin, R. Aniceto, A. Balakrishna, G. Balke, K. Black, G. Bokinsky, S. Cao, T. Charbonnier, V. Choudhary, F. Collins, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, M. Dhaka, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Habeeb, H. Hancock, K. Hausman, G. Hussein, V. Hwang, B. Ichter, C. Jacobsen, S. Jakubczak, R. Jen, T. Jones, G. Kammerer, B. Katz, L. Ke, M. Khadikov, C. Kuchi, M. Lamb, D. LeBlanc, B. LeCount, S. Levine, X. Li, A. Li-Bell, V. Lialin, Z. Liang, W. Lim, Y. Lu, E. Luo, V. Mano, N. Marwaha, A. Mongush, L. Murphy, S. Nair, T. Patterson, K. Pertsch, A. Z. Ren, G. Schelske, C. Sharma, B. Shi, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, J. Tang, J. Tanner, S. Tekeste, M. Torne, K. Vedder, Q. Vuong, A. Walling, H. Wang, J. Wang, X. Wang, C. Whalen, S. Whitmore, B. Williams, C. Xu, S. Yoo, L. Yu, W. Zhang, Z. Zhang, and U. Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. Technical report, Physical Intelligence, 2026. URL https://pi.website/pi07.

[11] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Hancock, K. Hausman, G. Hussein, B. Ichter, S. Jakubczak, R. Jen, T. Jones, B. Katz, L. Ke, C. Kuchi, M. Lamb, D. LeBlanc, S. Levine, A. Li-Bell, Y. Lu, V. Mano, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, C. Sharma, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, A. Swerdlow, J. Tanner, M. Torne, Q. Vuong, A. Walling, H. Wang, B. Williams, S. Yoo, L. Yu, U. Zhilinsky, and Z. Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025. URL https://arxiv.org/abs/2511.14759.

[12] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels. *arXiv preprint arXiv:2503.22634*, 2025.

[13] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.

[14] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, E. Byrne, Z. Chavis, J. Chen, F. Cheng, F.-J. Chu, S. Crane, A. Dasgupta, J. Dong, M. Escobar, C. Forigua, A. Gebreselasie, S. Haresh, J. Huang, M. M. Islam, S. Jain, R. Khirodkar, D. Kukreja, K. J. Liang, J.-W. Liu, S. Majumder, Y. Mao, M. Martin, E. Mavroudi, T. Nagarajan, F. Ragusa, S. K. Ramakrishnan, L. Seminara, A. Somayazulu, Y. Song, S. Su, Z. Xue, E. Zhang, J. Zhang, A. Castillo, C. Chen, X. Fu, R. Furuta, C. Gonzalez, P. Gupta, J. Hu, Y. Huang, Y. Huang, W. Khoo, A. Kumar, R. Kuo, S. Lakhavani, M. Liu, M. Luo, Z. Luo, B. Meredith, A. Miller, O. Oguntola, X. Pan, P. Peng, S. Pramanick, M. Ramazanova, F. Ryan, W. Shan, K. Somasundaram, C. Song, A. Southerland, M. Tateno, H. Wang, Y. Wang, T. Yagi, M. Yan, X. Yang, Z. Yu, S. C. Zha, C. Zhao, Z. Zhao, Z. Zhu, J. Zhuo, P. Arbelaez, G. Bertasius, D. Crandall, D. Damen, J. Engel, G. M. Farinella, A. Furnari, B. Ghanem, J. Hoffman, C. V. Jawahar, R. Newcombe, H. S. Park, J. M. Rehg,

Y. Sato, M. Savva, J. Shi, M. Z. Shou, and M. Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. URL https://arxiv.org/abs/2311.18259.

[15] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, Q. Ma, S. Nah, L. Magne, J. Xiang, Y. Xie, R. Zheng, D. Niu, Y. L. Tan, K. R. Zentner, G. Kurian, S. Indupuru, P. Jannaty, J. Gu, J. Zhang, J. Malik, P. Abbeel, M.-Y. Liu, Y. Zhu, J. Jang, and L. J. Fan. Dreamdojo: A generalist robot world model from large-scale human videos, 2026. URL https://arxiv.org/abs/2602.06949.

[16] J. Hejna, C. Bhateja, Y. Jiang, K. Pertsch, and D. Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning, 2024. URL https://arxiv.org/abs/2408.14037.

[17] Z. Zhang, J. Pang, Z. Yang, K. Li, M. Liao, S. Zhang, G. Chi, J. Guo, H. ang Gao, M. Shi, D. Ge, Y. Mu, J. Gu, R. Chen, H. Dong, H. Xu, L. Yi, Y. Zhu, H. Zhao, P. Wang, S. Zhang, G. Yao, J. Chen, H. Li, and H. Zhao. Dexora: Open-source vla for high-dof bimanual dexterity, 2026. URL https://arxiv.org/abs/2605.18722.

[18] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control, 2025. URL https://arxiv.org/abs/2502.05855.

[19] G. Daras, A. Rodriguez-Munoz, A. Klivans, A. Torralba, and C. C. Daskalakis. Ambient diffusion omni: Training good models with bad data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=MVYz4GmcUH.

[20] V. Tangkaratt, N. Charoenphakdee, and M. Sugiyama. Robust imitation learning from noisy demonstrations, 2021. URL https://arxiv.org/abs/2010.10181.

[21] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14 (3):391–412, 2003.

[22] A. Lukoianov, C. Yuan, J. Solomon, and V. Sitzmann. Locality in image diffusion models emerges from data statistics, 2025. URL https://arxiv.org/abs/2509.09672.

[23] 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. URL https://arxiv.org/abs/2303.04137.

[24] G. Daras, A. Dimakis, and C. C. Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PlVjIGaFdH.

[25] G. Daras, J. Ouyang-Zhang, K. Ravishankar, C. C. Daskalakis, A. Klivans, and D. J. Diaz. Ambient proteins - training diffusion models on noisy structures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Xll01vw606.

[26] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

[27] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

[28] J. Hejna, S. Mirchandani, A. Balakrishna, A. Xie, A. Wahid, J. Tompson, P. Sanketi, D. Shah, C. Devin, and D. Sadigh. Robot data curation with mutual information estimators, 2025. URL https://arxiv.org/abs/2502.08623.

[29] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. Liang, Q. V. Le, T. Ma, and A. W. Yu. Doremi: Optimizing data mixtures speeds up language model pretraining, 2023. URL https://arxiv.org/abs/2305.10429.

[30] C. Agia, R. Sinha, J. Yang, R. Antonova, M. Pavone, H. Nishimura, M. Itkina, and J. Bohg. Cupid: Curating data your robot loves with influence functions, 2025. URL https://arxiv.org/abs/2506.19121.

[31] S. Goyal, P. Maini, Z. C. Lipton, A. Raghunathan, and J. Z. Kolter. Scaling laws for data filtering– data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22702–22711, June 2024.

[32] J. Li, A. Fang, G. Smyrnis, M. Ivgi, M. Jordan, S. Gadre, H. Bansal, E. Guha, S. Keh, K. Arora, S. Garg, R. Xin, N. Muennighoff, R. Heckel, J. Mercat, M. Chen, S. Gururangan, M. Wortsman, A. Albalak, Y. Bitton, M. Nezhurina, A. Abbas, C.-Y. Hsieh, D. Ghosh, J. Gardner, M. Kilian, H. Zhang, R. Shao, S. Pratt, S. Sanyal, G. Ilharco, G. Daras, K. Marathe, A. Gokaslan, J. Zhang, K. Chandu, T. Nguyen, I. Vasiljevic, S. Kakade, S. Song, S. Sanghavi, F. Faghri, S. Oh, L. Zettlemoyer, K. Lo, A. El-Nouby, H. Pouransari, A. Toshev, S. Wang, D. Groeneveld, L. Soldaini, P. W. Koh, J. Jitsev, T. Kollar, A. G. Dimakis, Y. Carmon, A. Dave, L. Schmidt, and V. Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2025. URL https://arxiv.org/abs/2406.11794.

[33] A. Aali, M. Arvinte, S. Kumar, and J. I. Tamir. Solving inverse problems with score-based generative priors learned from noisy data. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, page 837–843. IEEE, Oct. 2023. doi: 10.1109/ieeeconf59524.2023.10477042. URL http://dx.doi.org/10.1109/IEEECONF59524.2023.10477042.

[34] T. Chen, Y. Zhang, Z. Wang, Y. N. Wu, O. Leong, and M. Zhou. Denoising score distillation: From noisy diffusion pretraining to one-step high-quality generation, 2025. URL https://arxiv.org/abs/2503.07578.

[35] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio. A survey on diffusion models for inverse problems, 2024. URL https://arxiv.org/abs/2410.00083.

[36] K. Shah, A. Kalavasis, A. Klivans, and G. Daras. Does generation require memorization? creative diffusion mod-

els using ambient diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=GGPM0z3dhU.

[37] A. Rodríguez-Muñoz, W. Daspit, A. Klivans, A. Torralba, C. Daskalakis, and G. Daras. Ambient dataloops: Generative models for dataset refinement, 2026. URL https://arxiv.org/abs/2601.15417.

[38] H. Lu, Q. Wu, and Y. Yu. SFBD: A method for training diffusion models with noisy data. In *Frontiers in Probabilistic Inference: Learning meets Sampling Workshop at ICLR*, 2025. URL https://openreview.net/pdf/8a1f8db4471b46d787ac9e685c15009b4566b5ed.pdf.

[39] H. Lu, Y. Yu, and D. Lo. Sfbd-omni: Bridge models for lossy measurement restoration with limited clean samples, 2026. URL https://arxiv.org/abs/2512.17051.

[40] C. Y. Park, S. Shoushtari, H. An, and U. S. Kamilov. Measurement score-based diffusion model, 2025. URL https://arxiv.org/abs/2505.11853.

[41] C. Modi, J. Han, E. Vanden-Eijnden, and J. Bruna. Generative modeling from black-box corruptions via self-consistent stochastic interpolants, 2026. URL https://arxiv.org/abs/2512.10857.

[42] B. Kawar, N. Elata, T. Michaeli, and M. Elad. Gsure-based diffusion model training with corrupted data, 2024. URL https://arxiv.org/abs/2305.13128.

[43] Y. Matsuzaki, S. Uchida, and S. Takezaki. Score: Clean image generation from diffusion models trained on noisy images, 2026. URL https://arxiv.org/abs/2604.09436.

[44] A. Bora, E. Price, and A. G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hy7fDog0b.

[45] W. Bai, Y. Wang, W. Chen, and H. Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations, 2024. URL https://arxiv.org/abs/2407.01014.

[46] D. Hosseintabar, F. Chen, G. Daras, A. Torralba, and C. Daskalakis. Diffem: Learning from corrupted data with diffusion models via expectation maximization, 2025. URL https://arxiv.org/abs/2510.12691.

[47] F. Rozet, G. Andry, F. Lanusse, and G. Louppe. Learning diffusion priors from observations by expectation maximization, 2025. URL https://arxiv.org/abs/2405.13712.

[48] J. Lyu, K. Liu, X. Zhang, H. Liao, Y. Feng, W. Zhu, T. Shen, J. Chen, J. Zhang, Y. Dong, W. Cui, S. Qi, S. Wang, Y. Zheng, M. Yan, X. Shi, H. Li, D. Zhao, M.-Y. Liu, Z. Zhang, L. Yi, Y. Wang, and H. Wang. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion, 2026. URL https://arxiv.org/abs/2602.12215.

[49] S. Cheng, L. Ma, Z. Chen, A. Mandlekar, C. Garrett, and D. Xu. Generalizable domain adaptation for sim-and-real policy co-training, 2026. URL https://arxiv.org/abs/2509.18631.

[50] R. Punamiya, D. Patel, P. Aphiwetsa, P. Kuppili, L. Y. Zhu, S. Kareer, J. Hoffman, and D. Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data, 2025. URL https://arxiv.org/abs/2509.19626.

[51] L. Liu, Z. Tang, L. Li, and D. Luo. Robust imitation learning from corrupted demonstrations, 2022. URL https://arxiv.org/abs/2201.12594.

[52] M. A. Pace, P. Dan, C. Ning, A. Bhardwaj, A. Du, E. W. Duan, W.-C. Ma, and K. Kedia. X-diffusion: Training diffusion policies on cross-embodiment human demonstrations, 2025. URL https://arxiv.org/abs/2511.04671.

[53] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[54] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[55] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[56] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[57] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

[58] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning, 2023. URL https://arxiv.org/abs/2306.02437.

[59] P. Pestana, Z. Ma, J. Reiss, Á. Barbosa, and D. Black. Spectral characteristics of popular commercial recordings 1950-2010. 01 2013.

[60] S. Dieleman. Diffusion is spectral autoregression, 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

[61] D.F. Gray. *The Observation and Analysis of Stellar Photospheres*. Cambridge Univ. Press, third edition edition, Nov. 2005.

[62] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 4th edition, 2002.

[63] E. Bauer, E. Nava, and R. K. Katzschmann. Latent action diffusion for cross-embodiment manipulation, 2026. URL https://arxiv.org/abs/2506.14608.

[64] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[65] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake. Motion planning around obstacles with convex optimization. *Science Robotics*, 8(84):eadf7843, 2023. doi: 10.1126/scirobotics.adf7843. URL https://www.science.org/doi/abs/10.1126/scirobotics.adf7843.

[66] S. M. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998. URL https://api.semanticscholar.org/CorpusID:

14744621.

[67] L. Foland, T. Cohn, A. Wei, N. Pfaff, B. Chen, and R. Tedrake. How well do diffusion policies learn kinematic constraint manifolds?, 2025. URL https://arxiv.org/abs/2510.01404.

[68] M. Dalal, A. Mandlekar, C. Garrett, A. Handa, R. Salakhutdinov, and D. Fox. Imitating task and motion planning with visuomotor transformers, 2023. URL https://arxiv.org/abs/2305.16309.

[69] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip. Motion planning networks, 2019. URL https://arxiv.org/abs/1806.05767.

[70] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models, 2024. URL https://arxiv.org/abs/2308.01557.

[71] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak. Neural mp: A generalist neural motion planner, 2024. URL https://arxiv.org/abs/2409.05864.

[72] B. P. Graesdal, S. Y. C. Chia, T. Marcucci, S. Morozov, A. Amice, P. A. Parrilo, and R. Tedrake. Towards tight convex relaxations for contact-rich manipulation, 2024. URL https://arxiv.org/abs/2402.10312.

[73] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL https://arxiv.org/abs/2410.13126.

[74] Y. Hu, F. Lin, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation, 2025. URL https://arxiv.org/abs/2410.18647.

[75] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021. URL https://arxiv.org/abs/2108.03298.

[76] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, page 1–6. IEEE, 2018.

[77] Y. Polyanskiy and Y. Wu. *Information theory: From coding to learning*. Cambridge university press, 2025.

[78] J. Zhang, Z. Zhou, H. Li, Y. Lai, W. Xia, H. Song, Y. Gong, and J. Mei. Hydra-dp3: Frequency-aware right-sizing of 3d diffusion policies for visuomotor control, 2026. URL https://arxiv.org/abs/2605.01581.

[79] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning, 2020. URL https://arxiv.org/abs/2010.01083.

[80] Y. Wang, Z. Wang, M. Nakura, P. Bhowal, C.-L. Kuo, Y.-T. Chen, Z. Erickson, and D. Held. Articubot: Learning universal articulated object manipulation policy via large scale simulation, 2025. URL https://arxiv.org/abs/2503.03045.

[81] H. Zhao, Y. Qi, B. Hu, Y. Zhu, Z. Chen, H. Tian, X. Zhu, O. Howell, H. Huang, R. Walters, D. Wang, and R. Platt. Generalizable hierarchical skill learning via object-centric representation, 2025. URL https://arxiv.org/abs/2510.21121.

[82] N. Gireesh, Y. Ju, C. Xu, W. Liu, Y. Wan, and H. Wang. Hdflow: Hierarchical diffusion-flow planning for long-horizon tasks, 2026. URL https://arxiv.org/abs/2605.04525.

[83] R. Tedrake. *Robotic Manipulation*. 2024. URL http://manipulation.mit.edu.

[84] J. J. Kuffner and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[85] G. Izatt. *Capturing Distributions over Worlds for Robotics with Spatial Scene Grammars*. Ph.d. thesis, Massachusetts Institute of Technology, Cambridge, MA, May 2022.

[86] N. Pfaff, H. Dai, S. Zakharov, S. Iwase, and R. Tedrake. Steerable scene generation with post training and inference-time search, 2025. URL https://arxiv.org/abs/2505.04831.

[87] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[88] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[89] R. Geraerts and M. H. Overmars. Creating high-quality paths for motion planning. *The international journal of robotics research*, 26(8):845–863, 2007.

[90] R. Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL https://drake.mit.edu.

[91] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[92] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z.

Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, V. Guizilini, D. A. Herrera, M. Heo, K. Hsu, J. Hu, M. Z. Irshad, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[93] M. M. Hong, J. Zhang, A. Nagabandi, and A. Gupta. Tmrl: Diffusion timestep-modulated pretraining enables exploration for efficient policy finetuning, 2026. URL https://arxiv.org/abs/2605.12236.

[94] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

APPENDIX

APPENDIX: TABLE OF CONTENTS

We empirically show that robot data exhibits a *spectral power law*. This spectral structure is special for two reasons. First, it is not universal: it is absent in natural audio [59, 60], stellar spectra [61], and Poisson point processes [62]. Second, we show that it makes Ambient Diffusion effective: it implies a global-to-local hierarchy (Figure 3), fast contraction through noise (Appendix Theorem 1), and a locality property (Appendix Theorem 2). Hence, Ambient Diffusion is well-suited for robotics.

A. Spectral Power Law in Robotics

The power spectral density (PSD) of a signal measures its energy at each frequency component, f. The signal exhibits a spectral power law if its PSD decays approximately as $|f|^{-\alpha}$, where $\alpha > 0$. In other words, its energy is concentrated in low-frequency components. A formal definition extended to random vectors is presented in Appendix Definition 1. Figure 4a and Appendix Figure 6 show that **robot action data exhibits a spectral power law**.

Implications for Diffusion Policy [23]: When a spectral power law exists, Dieleman [60] showed that denoisers generate low-frequency features at high noise, and high-frequency features at low noise. Appendix Figure 5 illustrates that in robotics, low frequencies encode *global* features (e.g. the high-level plan), while high frequencies encode *local* features (e.g. grasping primitives and smoothness). Thus, Diffusion Policies exhibit a *global-to-local* hierarchy: they learn *global* planning at high noise and *local* motion primitives at low noise. Appendix G0a provides further experimental evidence.

This hierarchy unlocks a new design axis for co-training algorithms in robotics: *noise-dependent data usage*. Diffusion Policies learn different features of the actions at each noise level; thus, suboptimal samples should only contribute to training at noise levels where p and q align.

Implications for Ambient Diffusion Policy: Due to the spectral power law, adding Gaussian noise to robot data acts as a high-frequency mask (Appendix Figure 5). Thus, if differences between p and q are concentrated in a high-frequency tail (i.e. low-level actions), then $t_{\min}$ will be small. We formalize this in Appendix Theorem 1. By only learning from suboptimal samples when $t > t_{\min}$, policies can learn the useful *global* features in $\mathcal{D}_q$ and ignore the *local* suboptimality (which is masked by noise).

Fortunately, the suboptimality in many robot datasets is in the local actions. For instance, non-expert and expert demonstrations may aim to achieve the same goal, but differ in the quality of the low-level motions. Similarly, in sim-to-real co-training or cross-embodied data, the high-level plan may be shared while the precise contact dynamics or feasible motions may differ.

B. Locality in Robotics

A denoiser exhibits "locality" if its output at each coordinate depends primarily on a small receptive field in the noisy input [22, 19].[1] In robotics, this means that each action output from the denoiser is most sensitive to its temporal neighbors in the noisy input. In Appendix Theorem 2, we prove that the spectral power law can imply locality in the optimal denoisers. Indeed, Figure 4b empirically identifies this phenomenon in robotics. Intuitively, this is because these denoisers focus on resolving *local* motion primitives, so they do not need to attend to distant actions in the input.

Now suppose that p and q share the same local motion primitives but differ globally. Due to locality, there exists a sufficiently small $t_{\max}$ such that for all $t < t_{\max}$, p and q agree within the receptive field of the optimal denoiser. Appendix B formalizes this statement. Section N leverages locality to learn *local* grasping primitives from data at low noise, even when its task-level structure is incorrect.

Remark: A spectral power law and locality exists in all the examined robot datasets (Appendix Table I). This suggests that these are general properties of robot data. Section IV provides methods for estimating $t_{\min}$ and $t_{\max}$ to leverage these properties in Ambient Diffusion Policy.

We prove two theorems that extend the theoretical foundations of Ambient Diffusion and rigorously justify our method for robotics. Prior works [19] empirically identify the spectral power law and locality as important properties. We prove that for a simplified model, the former implies both *fast contraction* through noise (Theorem 1) and *locality* of the optimal denoiser (Theorem 2). This establishes the spectral power law as the single fundamental property underlying the framework.

Both theorems are proved for zero-mean stationary Gaussians. We adopt this setting for two reasons: **1)** it simplifies the bounds, and **2)** it is a natural model for latent-space diffusion, which has been proposed in robotics [63], and is common in generative modeling more broadly [64]. Extending the analysis to general distributions is left to future work. Proofs are presented in Appendix D.

We begin by formally defining the spectral power law for general random vectors.

Definition 1 (Spectral Power Law)**.** Let X be a zero-mean, stationary random vector on $\mathbb{R}^N$, and let $S_X(f) := \mathbb{E}\big[|(\mathcal{F}X)(f)|^2\big]$ denote its power spectral density in the orthonormal Fourier basis. We say that the distribution of X follows a *spectral power law* if there exist a constant $C > 0$, a rolloff exponent $\alpha > 1$, and a cutoff frequency $f_0 \geq 1$ such that the high-frequency spectrum satisfies:

$$S_X(f) = C|f|^{-\alpha} \quad \text{for all } |f| \geq f_0.$$

The exact spectral behavior below the cutoff f_0 is permitted to be arbitrary but bounded.

C. Contraction Through Noise

Theorem 1 states that a spectral power law and low-frequency alignment between p and q imply *fast contraction*

[1] As in Lukoianov et al. [22], "locality" is a property of the data which is inherited by the optimal denoiser at low noise. We use the term "locality" loosely as both a property of the data and the resulting denoisers.

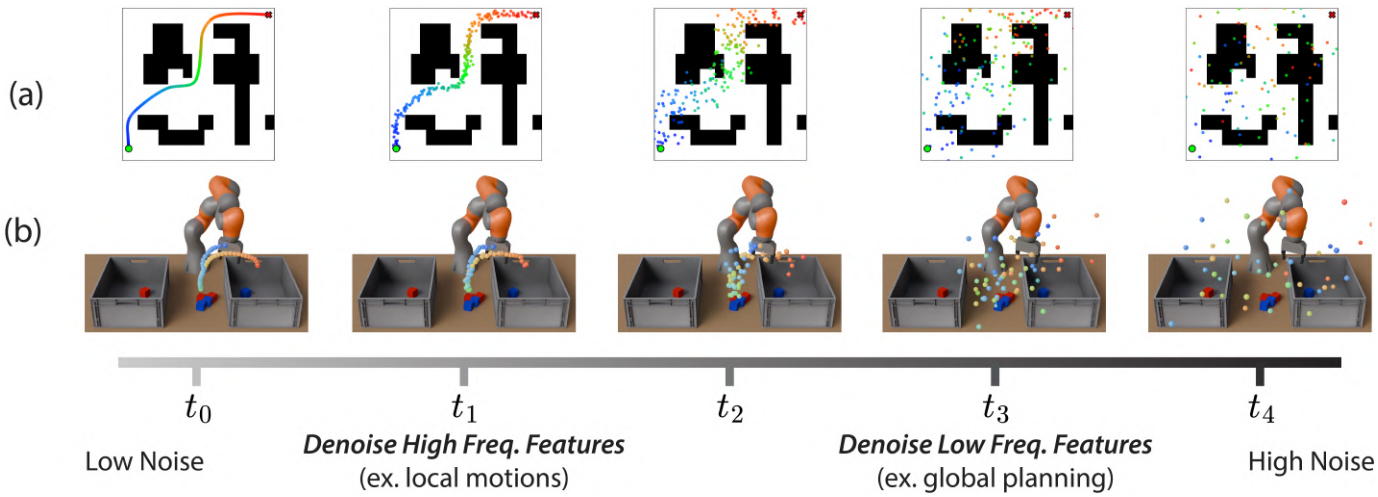
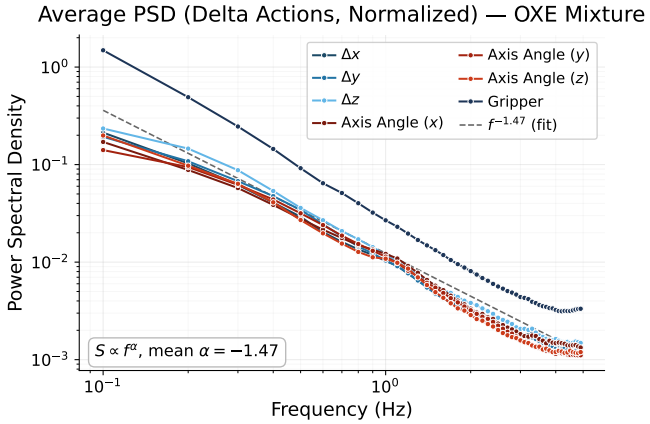
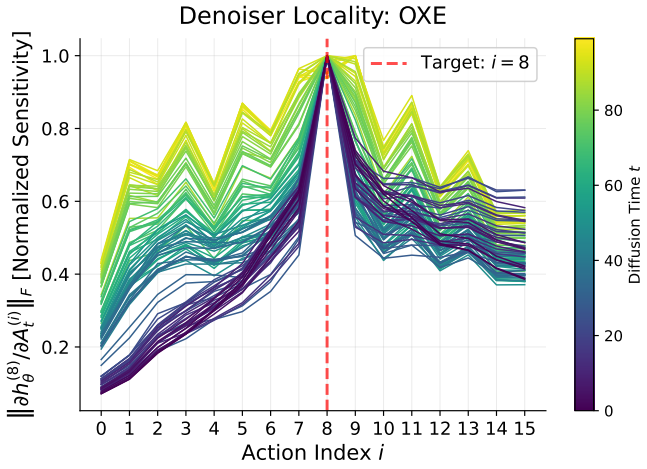


Fig. 3: **The spectral power law induces a *global*-to-*local* hierarchy in action diffusion.** This is analogous to the "coarse-to-fine" hierarchy in image diffusion [60]. **(a)** $t \geq t_2$ *(global planning):* the policy commits to a high-level path through the maze. $t < t_2$ *(local refinement):* it refines the plan to be collision-free at t_1 and smooth at t_0. **(b)** t_3: the policy plans to move a block to the right bin. t_2: it plans which block to grasp. $t < t_2$: it refines the grasping motion.



(a) **Spectral power law in robot action data.** The power spectral density for OXE [1], a collection of 2.4M robot episodes across 70+ datasets, follows a power law. Appendix Figure 6 shows that this spectral power law is a more general property of robot data.



(b) **Locality in robot action data.** Visualizing the sensitivity of the 8th action estimate to its neighboring actions (Eq. (77)). At low noise, the 8th action estimate is most sensitive to its temporal neighbors.

through noise. Moreover, this contraction is *independent of the original distance*.

Theorem 1 (Spectral Power Law implies fast contraction through noise)**.** *Let $p_0 = \mathcal{N}(0, \Sigma_p)$ and $q_0 = \mathcal{N}(0, \Sigma_q)$ be zero-mean stationary Gaussians on $\mathbb{R}^N$ with power spectra S_p, S_q. Fix a cutoff frequency $f^\star \geq 1$, an exponent $\alpha > \frac{1}{2}$, and a constant $C > 0$. Assume:*

1) ***(Low-frequency agreement.)*** *$S_p(f) = S_q(f)$ for every $f \leq f^\star$.*
2) ***(Power-law tail.)*** *$\max(S_p(f), S_q(f)) \leq C f^{-\alpha}$ for every f such that $f^\star < f \leq \lfloor N/2 \rfloor$.*

Let $p_t := p_0 * \mathcal{N}(0, \sigma_t^2 I)$ and $q_t := q_0 * \mathcal{N}(0, \sigma_t^2 I)$. If

$$\sigma_t^2 \geq C(f^\star)^{-\alpha}, \qquad (\star)$$

then

$$\boxed{\mathrm{d_{KL}}(p_t \| q_t) \leq \frac{4C^2}{(2\alpha - 1)\sigma_t^4 (f^\star)^{2\alpha - 1}}}$$

$$\boxed{\mathrm{d_{TV}}(p_t, q_t) \leq \frac{\sqrt{2}C}{\sigma_t^2 \sqrt{(2\alpha - 1)(f^\star)^{2\alpha - 1}}}}.$$

In contrast, the general TV-contraction bound (Section III) depends on the original TV distance (which can be arbitrarily close to 1) and decays only as $1/\sigma$. For our simple theoretical model, the spectral power law enables fast contraction and, consequently, high utilization of the suboptimal data in Ambient Diffusion.

D. Spectral Power Law Implies Locality

The next theorem shows that the spectral power law implies *locality* for the optimal denoiser at low noise levels.

Theorem 2 (Spectral Power Law implies locality of the optimal denoiser)**.** *Let $X_0 \sim \mathcal{N}(0, \Sigma)$ be a zero-mean stationary Gaussian on $\mathbb{R}^N$ whose power spectrum $S(f)$ follows a spectral power law with constant $C > 0$ and exponent $\alpha > 1$. Let $X_t = X_0 + \sigma_t Z$ be the variance-exploding noisy observation, and $h_t^\star(x) := \mathbb{E}[X_0 \mid X_t = x]$ denote the MMSE optimal denoiser.*

Let M_i be a circulant mask that sets to zero all coordinates of the input x at a circular distance strictly greater than L from index i. Then, for any mask size $1 \le L \le \lfloor N/2 \rfloor$, the absolute error in the i-th coordinate between the full optimal denoiser and the masked denoiser is bounded by:

$$\boxed{\left|(h_t^\star(x))_i - (h_t^\star(M_i x))_i\right| \le \frac{\alpha N \|x\|_\infty}{8L}.}$$

In short, at low noise levels the optimal denoiser is functionally myopic: it can reconstruct the signal by observing *only* a local spatial neighborhood, safely ignoring the global structure of the input. This is precisely the property that Ambient Diffusion Policy exploits to learn *local* action primitives from suboptimal data, even when its global structure is incorrect (Section N).

This appendix provides the proofs of the theoretical results stated in Appendix B: the spectral power law implies both *fast contraction* through noise (Theorem 1) and *locality* (Theorem 2) of the optimal denoiser. We then formally justify the classifier-based $t_{\min}$ annotation method from Phase 1 of our algorithm (Theorem 4). The Spectral Power Law (Definition 1) is stated in Appendix B, and the general TV-contraction bound is stated in the main body (Section III).

a) Preliminaries.: Given distributions P, Q, we recall the following information-theoretic divergences, used throughout this appendix:

$$\begin{aligned}
\mathrm{d_{TV}}(P,Q) &= \mathbb{E}_Q\left|1 - \frac{dP}{dQ}\right|, \\
\chi^2(P\|Q) &= \mathbb{E}_Q\left(\frac{dP}{dQ}\right)^2 - 1, \\
\mathrm{LC}(P,Q) &= 2\chi^2\left(P \,\middle\|\, \tfrac{1}{2}P + \tfrac{1}{2}Q\right).
\end{aligned}$$

Pinsker's inequality gives $\mathrm{d_{TV}}(P,Q)^2 \le \frac{1}{2}\mathrm{KL}(P,Q)$, and the LeCam inequality gives $\mathrm{d_{TV}}(P,Q)^2 \le \frac{1}{2}\mathrm{LC}(P,Q)$. We refer the reader to [77] for further background.

E. Spectral Power Law implies fast contraction through noise

Proof of Theorem 1: We start by making the observation that since the Discrete Fourier Transform (DFT) is an invertible operation, it holds that:

$$\mathrm{d_{KL}}(p_t\|q_t) = \mathrm{d_{KL}}(\mathcal{F}\#p_t\|\mathcal{F}\#q_t). \tag{1}$$

We denote with F the DFT-matrix. Multiplying with F vectors that are distributed according to a zero-mean Gaussian gives another zero-mean Gaussian distribution with covariance:

$$\begin{aligned}
\Sigma_{\mathcal{F}\#p_t} &= \mathbb{E}_{x\sim p_t}\left[(Fx)(Fx)^*\right] &(2)\\
&= F\mathbb{E}_{x\sim p_t}[xx^T]F^* &(3)\\
&= F\Sigma_{p_t}F^*. &(4)
\end{aligned}$$

We have that $\Sigma_{p_t} = \Sigma_p + \sigma_t^2 I$, where Σ_p is a circulant matrix (stationarity assumption). Hence, Σ_{p_t} is also a circulant matrix, and thus it can be diagonalized as follows:

$$\Sigma_{p_t} = F^*(\Lambda_p + \sigma_t^2 I)F. \tag{5}$$

Hence,

$$\begin{aligned}
\Sigma_{\mathcal{F}\#p_t} &= FF^*(\Lambda_p + \sigma_t^2 I)FF^* &(6)\\
&= \Lambda_p + \sigma_t^2 I. &(7)
\end{aligned}$$

Similarly, $\Sigma_{\mathcal{F}\#q_t} = \Lambda_q + \sigma_t^2 I$.

Using the formula for the KL distance between two zero-mean, diagonal, multivariate Gaussians, we get:

$$\mathrm{d_{KL}}(p_t\|q_t) \le \frac{1}{2}\sum_{i=0}^{N-1}\frac{(\Lambda_p)_{i,i}+\sigma_t^2}{(\Lambda_q)_{i,i}+\sigma_t^2} - \ln\left(\frac{(\Lambda_p)_{i,i}+\sigma_t^2}{(\Lambda_q)_{i,i}+\sigma_t^2}\right) - 1 \tag{8}$$

$$= \frac{1}{2}\sum_{f=0}^{N-1}\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2} - \ln\left(\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2}\right) - 1 \tag{9}$$

$$\le \sum_{f>f^\star}^{\lfloor N/2\rfloor}\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2} - \ln\left(\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2}\right) - 1, \tag{10}$$

where in the last inequality we used the fact that the spectrum is symmetric around the Nyquist frequency and that p and q agree on all frequencies up to $f^\star$.

Let's consider the function: $g(r) = r - \ln r - 1$. We will start by showing that $g(r) \le (r-1)^2$ for all $r \in [1/2, \infty)$.

Let

$$\begin{aligned}
h(r) &= g(r) - (r-1)^2 &(11)\\
&= r - \ln r - 1 - r^2 + 2r - 1 &(12)\\
&= -r^2 + 3r - \ln r - 2. &(13)
\end{aligned}$$

We hence have:

$$\begin{aligned}
h'(r) &= -2r + 3 - \frac{1}{r} &(14)\\
&= \frac{-2r^2 + 3r - 1}{r}. &(15)
\end{aligned}$$

The determinant of the nominator is $\Delta = 1$ and there are two points that make $h'(r) = 0$, which are:

$$r_1 = \frac{1}{2}, \quad r_2 = 1. \tag{16}$$

We have that $h'(r)$ is negative for $r > 1$ and for $r \in (0, 1/2)$. Hence, in $[1/2, \infty)$ the maximum of $h(r)$ is $h(1) = 0$. Hence, the function $h(r)$ is ≤ 0 for all $r \in [1/2, \infty)$.

We now want to get some bounds on the quantity $\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2}$. First, note that:

$$\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2} \geq \frac{\sigma_t^2}{\max(S_p(f),S_q(f))+\sigma_t^2}. \tag{17}$$

For all $\lfloor N/2 \rfloor \geq f > f^\star$, we further have (by Power-law tail assumption) that:

$$\max(S_p(f),S_q(f)) \leq Cf^{-\alpha}, \tag{18}$$

which gives that:

$$\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2} \geq \frac{\sigma_t^2}{Cf^{-\alpha}+\sigma_t^2}. \tag{19}$$

By assumption, we have that:

$$\sigma_t^2 \geq C\frac{1}{(f^\star)^\alpha}. \tag{20}$$

But:

$$\frac{C}{f^\alpha} < \frac{C}{(f^\star)^\alpha}, \quad \forall f \text{ such that } f^\star < f \leq \lfloor N/2 \rfloor. \tag{21}$$

Hence,

$$Cf^{-\alpha} < \sigma_t^2. \tag{22}$$

Plugging this into Eq. (19), we get:

$$\frac{S_p(f)+\sigma_t^2}{S_q(f)+\sigma_t^2} \geq 1/2. \tag{23}$$

By combining Eq. 23 and the fact that $g(r) \leq (r-1)^2$ for all $r \in [1/2, \infty)$, we can write Eq. (10) as follows:

$$\begin{aligned}
\mathrm{d_{KL}}(p_t\|q_t) &\leq \sum_{f>f^\star}^{\lfloor N/2 \rfloor} \left(\frac{S_p(f)-S_q(f)}{S_q(f)+\sigma_t^2}\right)^2 && (24)\\
&\leq \sum_{f>f^\star}^{\lfloor N/2 \rfloor} \left(\frac{2\max(S_p(f),S_q(f))}{\sigma_t^2}\right)^2 && (25)\\
&\leq \sum_{f>f^\star}^{\lfloor N/2 \rfloor} 4\frac{C^2 f^{-2\alpha}}{\sigma_t^4} && (26)\\
&= \frac{4C^2}{\sigma_t^4} \sum_{f>f^\star}^{\lfloor N/2 \rfloor} \left(\frac{1}{f}\right)^{2\alpha} && (27)\\
&\leq \frac{4C^2}{\sigma_t^4} \int_{f^\star}^{\infty} \left(\frac{1}{f}\right)^{2\alpha} \mathrm{d}f && (28)\\
&= \frac{4C^2}{\sigma_t^4(2\alpha-1)(f^\star)^{2\alpha-1}}. && (29)
\end{aligned}$$

The TV distance bound follows from a direct application of Pinsker's inequality. ∎

F. Spectral Power Law implies locality

Proof of Theorem 2: Since X_0 is distributed according to a Gaussian distribution, the MMSE optimal denoiser is a linear function, given as:

$$\mathbb{E}[X_0|X_t = x] = \Sigma(\Sigma+\sigma_t^2 I)^{-1}x. \tag{30}$$

From the stationarity assumption, Σ is a circulant matrix and so is the matrix $\Sigma+\sigma_t^2 I$. The inverse of a circulant matrix is another circulant matrix, and multiplying two circulant matrices gives another circulant matrix. Hence, the whole matrix $\Sigma(\Sigma+\sigma_t^2 I)^{-1}$ is circulant.

Multiplication with a circulant matrix can be written as a circulant convolution; hence, we can express the optimal denoiser in the following equivalent way:

$$(\mathbb{E}[X_0|X_t = x])_i = \sum_{j=0}^{N-1} k_t((i-j) \mod N)x_j, \tag{31}$$

where k_t is the first column of the matrix $\Sigma(\Sigma+\sigma_t^2 I)^{-1}$. Let us now analyze how the prediction of the optimal denoiser changes when, instead of feeding the full input, we input a masked version $M_i x$ of the input where M_i is a circulant mask.

We have that:

$$\begin{aligned}
\mathrm{error}_i &= \Bigg| \sum_{j=0}^{N-1} k_t((i-j) \mod N)x_j \\
&\quad - \sum_{j=0}^{N-1} k_t((i-j) \mod N)M_i x_j \Bigg| && (32)\\
&= \Bigg| \sum_{j:\mathrm{dist}(i,j)>L} k_t((i-j) \mod N)x_j \Bigg|, && (33)\\
&\leq ||x||_\infty \sum_{j:\mathrm{dist}(i,j)>L} \Big| k_t((i-j) \mod N) \Big|, && (34)
\end{aligned}$$

where $\mathrm{dist}(i,j) = \min(|i-j|, N-|i-j|)$. Let $m = (i-j) \mod N$. First note that $L+1 \leq m \leq \lfloor N/2 \rfloor$. Second, note that for each value m, there are two indices j that can give this value. Hence, we can rewrite the sum as:

$$\mathrm{error}_i \leq ||x||_\infty \sum_{m=L+1}^{\lfloor N/2 \rfloor} |k_t(m)| + |k_t(N-m)|. \tag{35}$$

We now need to analyze what the kernel k_t is. We have defined this vector as the first column of the matrix $\Sigma(\Sigma+\sigma_t^2 I)^{-1}$. Because the matrix is circulant, we have that $k_t(m) = k_t(N-m)$. Hence,

$$\mathrm{error}_i \leq 2||x||_\infty \sum_{m=L+1}^{\lfloor N/2 \rfloor} |k_t(m)|. \tag{36}$$

By definition:

$$k_t(m) = (\Sigma(\Sigma+\sigma_t^2 I)^{-1}e_1)(m). \tag{37}$$

The matrix $\Sigma(\Sigma + \sigma_t^2 I)^{-1}$ is circulant, and hence it is diagonalized by the DFT. Hence, we have:

$$k_t(m) = (F^* W F e_1)(m), \tag{38}$$

where W is the diagonal matrix with diagonal entries $W_{f,f} = \frac{S(f)}{S(f)+\sigma_t^2}$.

We observe now that $k_t(m)$ is exactly the Inverse Discrete Fourier Transform of the Wiener filter. Hence, $k_t(m)$ can be equivalently written as:

$$k_t(m) = \frac{1}{N} \sum_{f=0}^{N-1} W_{f,f} e^{i\frac{2\pi f m}{N}}. \tag{39}$$

Because the original signal X_0 is real-valued, its power spectrum and the resulting Wiener filter $W_f = W_{f,f}$ are symmetric around the Nyquist frequency, meaning $W_{N-f} = W_f$. This conjugate symmetry perfectly cancels the imaginary sine components, allowing us to write the kernel entirely in terms of real cosines:

$$k_t(m) = \frac{1}{N} \sum_{f=0}^{N-1} W_f \cos\left(\frac{2\pi f m}{N}\right). \tag{40}$$

To extract the spatial decay of this kernel as m increases, let $\omega_m = \frac{2\pi m}{N}$. We multiply both sides of the equation by the phase-shifting factor $2(1 - \cos\omega_m)$:

$$\begin{aligned} &2(1-\cos\omega_m)k_t(m) \\ &= \frac{1}{N}\sum_{f=0}^{N-1} W_f \left[2\cos(f\omega_m) - 2\cos(f\omega_m)\cos(\omega_m)\right]. \end{aligned} \tag{41}$$

Applying the standard trigonometric identity $2\cos(A)\cos(B) = \cos(A+B) + \cos(A-B)$ to the second term gives:

$$\begin{aligned} &2(1-\cos\omega_m)k_t(m) \\ &= \frac{1}{N}\sum_{f=0}^{N-1} W_f \Big[2\cos(f\omega_m) - \cos((f+1)\omega_m) \\ &\qquad - \cos((f-1)\omega_m)\Big]. \end{aligned} \tag{42}$$

Because the discrete spectrum is periodic ($W_N = W_0$) and the cosine function is periodic over N, we can shift the indices of summation for the $(f+1)$ and $(f-1)$ terms to group them by the common $\cos(f\omega_m)$ factor. This perfectly regroups the sum into the central second difference $\Delta^2 W_f = W_{f+1} - 2W_f + W_{f-1}$:

$$2(1-\cos\omega_m)k_t(m) = -\frac{1}{N}\sum_{f=0}^{N-1} (\Delta^2 W_f)\cos(f\omega_m). \tag{43}$$

We now take the absolute value of both sides. For the left side, we apply the half-angle trigonometric identity $2(1 - \cos\omega_m) = 4\sin^2\left(\frac{\omega_m}{2}\right) = 4\sin^2\left(\frac{\pi m}{N}\right)$. Since $|\cos(f\omega_m)| \leq 1$, it drops out of the right side:

$$4\sin^2\left(\frac{\pi m}{N}\right)|k_t(m)| \leq \frac{1}{N}\sum_{f=0}^{N-1} |\Delta^2 W_f|. \tag{44}$$

Crucially, the assumption that $S(f)$ follows a spectral power law dictates the structural behavior of this Wiener filter. For high frequencies $f \geq 1$, the power law $S(f) = Cf^{-\alpha}$ ensures that W_f monotonically and smoothly decays to zero. We can bound the discrete slope of this filter by analyzing its continuous extension $W(f) = \frac{C}{C+\sigma_t^2 f^\alpha}$. Its derivative is:

$$W'(f) = \frac{-\alpha C \sigma_t^2 f^{\alpha-1}}{(C + \sigma_t^2 f^\alpha)^2}. \tag{45}$$

By the Arithmetic Mean - Geometric Mean inequality, $(C + \sigma_t^2 f^\alpha)^2 \geq 4C\sigma_t^2 f^\alpha$. Substituting this into the derivative gives a strict bound on the slope:

$$|W'(f)| \leq \frac{\alpha C \sigma_t^2 f^{\alpha-1}}{4C\sigma_t^2 f^\alpha} = \frac{\alpha}{4f} \leq \frac{\alpha}{4}. \tag{46}$$

The sequence of discrete first differences ΔW_f samples this continuous slope. Over the positive half of the spectrum, the slope starts near zero, reaches a maximum magnitude bounded by $\alpha/4$, and gradually returns to zero. Because the function is convex in the tail, the second differences maintain a constant sign, allowing the absolute sum to telescope perfectly. The total absolute second variation for the positive half of the spectrum is therefore exactly bounded by twice the maximum slope, $2(\alpha/4) = \alpha/2$. Due to the conjugate symmetry of the spectrum, the negative frequencies contribute an identical amount.

Therefore, the total absolute second variation over the entire spectrum is strictly bounded by the power-law exponent:

$$\sum_{f=0}^{N-1} |\Delta^2 W_f| \leq \alpha. \tag{47}$$

Substituting this variation bound back into our inequality yields:

$$4\sin^2\left(\frac{\pi m}{N}\right)|k_t(m)| \leq \frac{\alpha}{N}. \tag{48}$$

For the spatial domain $m \leq \lfloor N/2 \rfloor$, the argument $\frac{\pi m}{N}$ falls within the interval $[0, \pi/2]$. In this domain, we can apply the linear lower bound for the sine function, $\sin(x) \geq \frac{2x}{\pi}$, which gives $\sin^2\left(\frac{\pi m}{N}\right) \geq \frac{4m^2}{N^2}$. Substituting this yields:

$$4\left(\frac{4m^2}{N^2}\right)|k_t(m)| \leq \frac{\alpha}{N} \implies |k_t(m)| \leq \frac{\alpha N}{16m^2}. \tag{49}$$

We have thus proven that the optimal denoiser kernel decays spatially at a rate of $\mathcal{O}(1/m^2)$. Returning to the error of the masked denoiser, we substitute this polynomial decay bound into the error sum:

$$\begin{aligned} \text{error}_i &\leq 2\|x\|_\infty \sum_{m=L+1}^{\lfloor N/2 \rfloor} |k_t(m)| \\ &\leq \frac{\alpha N \|x\|_\infty}{8} \sum_{m=L+1}^{\lfloor N/2 \rfloor} \frac{1}{m^2}. \end{aligned} \tag{50}$$

We can upper-bound this remaining discrete sum by the corresponding continuous integral over the spatial tail:

$$\sum_{m=L+1}^{\lfloor N/2 \rfloor} \frac{1}{m^2} \leq \int_{L}^{\infty} \frac{1}{y^2} \mathrm{d}y = \frac{1}{L}. \tag{51}$$

Hence, the approximation error of the masked denoiser is strictly bounded by:

$$\mathrm{error}_i \leq \frac{\alpha N \|x\|_\infty}{8L}. \tag{52}$$

This demonstrates that as the mask size L increases, the error drops off linearly, mathematically proving that the spectral power law forces the MMSE optimal denoiser to be spatially local. ∎

G. Theoretical Justification of the Classifier-Based Annotation

We prove that the classifier-based $t_{\min}$ annotation method introduced in Phase 1 of Section IV comes with a formal guarantee: suppose the classifier, f_θ,[2] is ε-optimal and the threshold is τ. Then annotating $t_{\min}$ using the method described in Section IV bounds the LeCam distance between the noisy distributions p_t and q_t (Theorem 4). Simply put, it guarantees that p_t and q_t are similar (i.e. contracted) for $t > t_{\min}$.

a) Classifier threshold implies distributional closeness.: Consider the population cross-entropy loss

$$\mathcal{L}_{\mathrm{CE}}(f) = -\mathbb{E}_{c,x}[(1-c)\log(1-f(x)) + c \log f(x)],$$

where we first sample $c \sim \mathrm{Bern}(1/2)$ and then sample $x \sim P$ if $c = 1$ or $x \sim Q$ otherwise. Let $M = \frac{1}{2}P + \frac{1}{2}Q$ denote the mixture. The optimal classifier is

$$f^* = \arg\min_f \mathcal{L}_{\mathrm{CE}}(f), \quad f^*(x) = \frac{dP}{dP + dQ}(x) = \frac{1}{2}\frac{dP}{dM}(x). \tag{53}$$

Theorem 3 (Optimal classifier threshold implies distributional closeness)**.** *Let f^* be the optimal classifier in Eq.* (53)*. Then*

$$\mathbb{E}_Q f^*(x) = \frac{1}{2} - \frac{1}{2}\mathrm{LC}(P,Q).$$

In particular, $\mathbb{E}_Q f^*(x) \geq \frac{1}{2} - \tau$ implies $\mathrm{LC}(P,Q) \leq 2\tau$.

Proof: We compute

$$\mathbb{E}_Q f^* = \int \frac{dP\, dQ}{dP + dQ} = \frac{1}{2}\int \frac{(dP+dQ)^2 - dP^2 - dQ^2}{dP + dQ}.$$

Then

$$\mathbb{E}_Q f^* = 1 - \frac{1}{2}\int \frac{dP^2}{dP+dQ} - \frac{1}{2}\int \frac{dQ^2}{dP+dQ}.$$

Noting $\int \frac{dP^2}{dP+dQ} = \frac{1}{2}(\chi^2(P\|M) + 1)$ gives

$$\mathbb{E}_Q f^* = \frac{1}{2} - \frac{1}{2}\chi^2(P\|M) - \frac{1}{2}\chi^2(Q\|M).$$

[2]Section IV uses c_ϕ to denote the classifier. We switched notation to f_θ for the proofs since they use c to denote a Bernoulli random variable.

Using $\mathrm{LC}(P,Q) = 2\chi^2(P\|M) = 2\chi^2(Q\|M)$, we conclude

$$\mathbb{E}_Q f^* = \frac{1}{2} - \frac{1}{2}\mathrm{LC}(P,Q).$$

The threshold condition $\mathbb{E}_Q f^* \geq \frac{1}{2} - \tau$ then immediately gives $\mathrm{LC}(P,Q) \leq 2\tau$. ∎

We now extend this result to approximate classifiers trained on the cross-entropy loss.

Theorem 4 (Approximate classifier threshold implies distributional closeness)**.** *Suppose $f_\theta : \mathbb{R}^d \to [0,1]$ is an ε-optimal classifier, i.e. $\mathcal{L}_{\mathrm{CE}}(f_\theta) \leq \mathcal{L}_{\mathrm{CE}}(f^*) + \varepsilon$. Then $\mathbb{E}_Q f_\theta(x) \geq \frac{1}{2} - \tau$ implies $\mathrm{LC}(P,Q) \leq 2\tau + 2\sqrt{\varepsilon}$.*

Proof: The cross-entropy loss admits the decomposition

$$\mathcal{L}_{\mathrm{CE}}(f) = \mathbb{E}_{x\sim M}\, d(f^*(x)\|f(x)) + \mathrm{const},$$

where $d(p\|q) = \mathrm{KL}(\mathrm{Bern}(p)\|\mathrm{Bern}(q))$ is the binary KL divergence. The ε-accuracy condition implies $\mathbb{E}_{x\sim M} d(f^*(x)\|f(x)) \leq \varepsilon$, and by non-negativity $\mathbb{E}_{x\sim Q} d(f^*(x)\|f(x)) \leq 2\varepsilon$. Applying Pinsker's inequality and $\mathrm{d_{TV}}(p,q) = |p-q|$ for Bernoulli variables yields

$$\mathbb{E}_{x\sim Q}(f_\theta(x) - f^*(x))^2 \leq \varepsilon.$$

By Jensen, $\mathbb{E}_{x\sim Q} f^*(x) \geq \mathbb{E}_{x\sim Q} f_\theta(x) - \sqrt{\varepsilon}$, so

$$\mathbb{E}_{x\sim Q} f^*(x) \geq \frac{1}{2} - \tau - \sqrt{\varepsilon}.$$

Applying Theorem 3 recovers the claim. ∎

The paper presents Ambient Diffusion Policy from the variance-exploding perspective of diffusion [55] to lighten notation. Our implementation builds off the original Diffusion Policy [23] work, which uses a variance-preserving implementation. We first show that the two perspectives are equivalent up to a change of variables.

The *variance exploding* (VE) and *variance preserving* (VP) forward processes are defined in Eq. (54) and (55) respectively [55]:

$$X_t = X_0 + \sigma_{\mathrm{VE}}(t)\, Z, \tag{54}$$

$$X_t = \sqrt{1 - \sigma^2_{\mathrm{VP}}(t)}\, X_0 + \sigma_{\mathrm{VP}}(t)\, Z, \tag{55}$$

where $Z \sim \mathcal{N}(0, I)$, $\sigma_{\mathrm{VE}}(t)$ is an unbounded increasing function with $\sigma_{\mathrm{VE}}(0) = 0$, and $\sigma_{\mathrm{VP}}(t)$ is an increasing function with $\sigma_{\mathrm{VP}}(0) = 0$ and $\sigma_{\mathrm{VP}}(T) = 1$. Without loss of generality, we assume the data is normalized so that $\mathrm{Var}[X_0] = 1$.

Proposition 1. *The VP and VE processes are equivalent up to a time-dependent rescaling of space.*

Proof: Given a VE schedule $\sigma_{\mathrm{VE}}(t)$, define

$$\sigma_{\mathrm{VP}}(t) := \frac{\sigma_{\mathrm{VE}}(t)}{\sqrt{1 + \sigma^2_{\mathrm{VE}}(t)}}, \qquad c(t) := \sqrt{1 + \sigma^2_{\mathrm{VE}}(t)}. \tag{56}$$

Under this mapping,

$$\mathrm{SNR}_{\mathrm{VE}}(t) = \frac{1}{\sigma^2_{\mathrm{VE}}(t)} = \frac{1 - \sigma^2_{\mathrm{VP}}(t)}{\sigma^2_{\mathrm{VP}}(t)} = \mathrm{SNR}_{\mathrm{VP}}(t), \tag{57}$$

so both processes have the same SNR at every t. The rescaled VE variable, $\widetilde{X}_t$, satisfies

$$\widetilde{X}_t := \frac{X_t^{\mathrm{VE}}}{c(t)} = X_t^{\mathrm{VP}}. \tag{58}$$

Thus, the VP process is the VE process in normalized coordinates $x_t^{\mathrm{VE}} \mapsto x_t^{\mathrm{VE}}/c(t) = x_t^{\mathrm{VP}}$. ∎

Proposition 2. *The score functions of the VP and VE processes are related by:*

$$s^{\mathrm{VP}}(\widetilde{x}_t, t) = c(t) \cdot s^{\mathrm{VE}}(c(t)\,\widetilde{x}_t,\, t)\,. \tag{59}$$

Proof: Let p_t^{VP} and p_t^{VE} denote the marginal densities of X_t^{VP} and X_t^{VE} respectively. Given the change of variables $x_t^{\mathrm{VE}} = c(t)\,x_t^{\mathrm{VP}}$, we have that $p_t^{\mathrm{VP}}(x_t^{\mathrm{VP}}) = c(t)^d\, p_t^{\mathrm{VE}}(c(t)\,x_t^{\mathrm{VP}})$, where d is the dimension. Taking the gradient with respect to x_t^{VP}:

$$\begin{aligned} s^{\mathrm{VP}}(x_t^{\mathrm{VP}}, t) &= \nabla_{x_t^{\mathrm{VP}}} \log p_t^{\mathrm{VP}}(x_t^{\mathrm{VP}}) \\ &= \nabla_{x_t^{\mathrm{VP}}} \log p_t^{\mathrm{VE}}(c(t)\,x_t^{\mathrm{VP}}) + \cancelto{0}{\nabla_{x_t^{\mathrm{VP}}} \log c(t)^d} \\ &= c(t) \cdot s^{\mathrm{VE}}\big(c(t)\,x_t^{\mathrm{VP}},\, t\big)\,, \end{aligned} \tag{60}$$

where the last line follows from the chain rule. ∎

Proposition 3. *Let h^*_{VE} and h^*_{VP} denote the optimal denoisers for the VE and VP processes respectively. Then:*

$$h^*_{\mathrm{VP}}(x_t^{\mathrm{VP}}, t) = h^*_{\mathrm{VE}}\big(c(t)\,x_t^{\mathrm{VP}},\, t\big)\,. \tag{61}$$

Proof: By Tweedie's formula, the optimal denoisers are related to the scores via:

$$h^*_{\mathrm{VE}}(x_t^{\mathrm{VE}}, t) = x_t^{\mathrm{VE}} + \sigma^2_{\mathrm{VE}}(t)\, s^{\mathrm{VE}}(x_t^{\mathrm{VE}}, t), \tag{62}$$

$$h^*_{\mathrm{VP}}(x_t^{\mathrm{VP}}, t) = \frac{x_t^{\mathrm{VP}} + \sigma^2_{\mathrm{VP}}(t)\, s^{\mathrm{VP}}(x_t^{\mathrm{VP}}, t)}{\sqrt{1 - \sigma^2_{\mathrm{VP}}(t)}}. \tag{63}$$

Substituting the score correspondence from Proposition 2 and the definition of $c(t)$ in (56) yields the desired result. ∎

Therefore, any result derived in one framework translates to the other via $c(t) = \sqrt{1 + \sigma^2_{\mathrm{VE}}(t)}$.

In the original Ambient Diffusion work, Daras et al. [24] propose an alternative loss function, known as the Ambient loss. Unlike the denoising loss from DDPM [54], the Ambient loss assumes no access to the clean samples, x_0, and instead uses $x_{t_{\min}}$ as the prediction target. Nonetheless, it is minimized by the same denoiser as the DDPM loss. In the image domain, the Ambient loss has been shown to improve generalization. Daras et al. [24] derive the Ambient loss for VE diffusion. In this section, we derive it for VP diffusion.

Proposition 4. *The ϵ-prediction VP Ambient loss for $t > t_{\min}$ is given by*

$$\begin{aligned} \mathcal{L}(\theta) = \mathbb{E}\Big[\big\|\epsilon_\theta(x_t, t) &- \tfrac{\sigma_t(1-\sigma^2_{t_{\min}})}{\sigma_t^2 - \sigma^2_{t_{\min}}}\, x_t \\ &+ \tfrac{\sigma_t\sqrt{(1-\sigma_t^2)(1-\sigma^2_{t_{\min}})}}{\sigma_t^2 - \sigma^2_{t_{\min}}}\, x_{t_{\min}}\big\|_2^2\Big], \end{aligned} \tag{64}$$

and minimized by $\epsilon^*_\theta = \mathbb{E}[Z \mid X_t = x_t]$.

Proof: Let X_t and Z be the random variables from the forward process in VP diffusion, Eq. (55). Then:

$$\mathbb{E}[Z \mid X_t = x_t] = \frac{x_t - \sqrt{1-\sigma_t^2}\,\mathbb{E}[X_0 \mid X_t = x_t]}{\sigma_t}. \tag{65}$$

By the double Tweedie identity from Daras et al. [24]:

$$\begin{aligned} \mathbb{E}[X_0 \mid X_t = x_t] = \tfrac{\sigma_t^2\sqrt{1-\sigma^2_{t_{\min}}}}{\sigma_t^2 - \sigma^2_{t_{\min}}}\mathbb{E}[X_{t_{\min}} \mid X_t = x_t] \\ - \tfrac{\sigma^2_{t_{\min}}\sqrt{1-\sigma_t^2}}{\sigma_t^2 - \sigma^2_{t_{\min}}} x_t. \end{aligned} \tag{66}$$

Substituting Eq. (66) into Eq. (65) and rearranging gives:

$$\mathbb{E}[X_{t_{\min}} \mid X_t = x_t] = \alpha_t\, \mathbb{E}[Z \mid X_t = x_t] + \beta_t\, x_t, \tag{67}$$

$$\alpha_t := -\frac{\sigma_t^2 - \sigma^2_{t_{\min}}}{\sigma_t\sqrt{(1-\sigma_t^2)(1-\sigma^2_{t_{\min}})}}, \tag{68}$$

$$\beta_t := \frac{1 - \sigma^2_{t_{\min}}}{\sqrt{(1-\sigma_t^2)(1-\sigma^2_{t_{\min}})}}. \tag{69}$$

Define $g_\theta(x_t, t) := \alpha_t\, \epsilon_\theta(x_t, t) + \beta_t\, x_t$. Since $t > t_{\min}$, we have $\alpha_t \neq 0$ and the map $\epsilon_\theta \mapsto g_\theta$ is a bijection. Thus, minimizing

$$\mathbb{E}\big[\|g_\theta(x_t, t) - x_{t_{\min}}\|^2\big] \tag{70}$$

over g_θ is equivalent to minimizing over ϵ_θ. The minimizer of Eq. (70) is $g^*_\theta = \mathbb{E}[X_{t_{\min}} \mid X_t = x_t]$, which by Eq. (67) implies $\epsilon^*_\theta = \mathbb{E}[Z \mid X_t = x_t]$. Rescaling the loss in Eq. (70) by α_t^2 yields:

$$\mathbb{E}\left[\left\|\epsilon_\theta(x_t, t) + \frac{\beta_t}{\alpha_t}x_t - \frac{1}{\alpha_t}x_{t_{\min}}\right\|_2^2\right], \tag{71}$$

which expands to the loss in the proposition. ∎

Proposition 5. *The x_0-prediction VP Ambient loss for $t > t_{\min}$ is given by*

$$\mathbb{E}\left[\left\|h_\theta(x_t, t) + \frac{\sigma^2_{t_{\min}}\sqrt{1-\sigma_t^2}}{\sigma_t^2 - \sigma^2_{t_{\min}}}x_t - \frac{\sigma_t^2\sqrt{1-\sigma^2_{t_{\min}}}}{\sigma_t^2 - \sigma^2_{t_{\min}}}x_{t_{\min}}\right\|_2^2\right], \tag{72}$$

and minimized by $h^*_\theta = \mathbb{E}[X_0 \mid X_t = x_t]$.

Proof: Isolating $\mathbb{E}[X_{t_{\min}} \mid X_t = x_t]$ in the double Tweedie identity Eq. (66) gives:

$$\mathbb{E}[X_{t_{\min}} \mid X_t = x_t] = \tilde{\alpha}_t\, \mathbb{E}[X_0 \mid X_t = x_t] + \tilde{\beta}_t\, x_t, \tag{73}$$

$$\tilde{\alpha}_t := \tfrac{\sigma_t^2 - \sigma^2_{t_{\min}}}{\sigma_t^2\sqrt{1-\sigma^2_{t_{\min}}}}, \quad \tilde{\beta}_t := \tfrac{\sigma^2_{t_{\min}}\sqrt{1-\sigma_t^2}}{\sigma_t^2\sqrt{1-\sigma^2_{t_{\min}}}}. \tag{74}$$

Define $\tilde{g}_\theta(x_t, t) := \tilde{\alpha}_t\, h_\theta(x_t, t) + \tilde{\beta}_t\, x_t$. Since $t > t_{\min}$, then $\tilde{\alpha}_t \neq 0$ and the map $h_\theta \mapsto \tilde{g}_\theta$ is a bijection. Thus, minimizing

$$\mathbb{E}\big[\|\tilde{g}_\theta(x_t, t) - x_{t_{\min}}\|^2\big] \tag{75}$$

over $\tilde{g}_\theta$ is equivalent to minimizing over h_θ. The minimizer of Eq. (75) is $\tilde{g}^*_\theta = \mathbb{E}[X_{t_{\min}} \mid X_t = x_t]$, which by Eq. (73)

implies $h_\theta^* = \mathbb{E}[X_0 \mid X_t = x_t]$. Rescaling the loss in Eq. (75) by $\tilde{\alpha}_t^2$ yields:

$$\mathbb{E}\left[\left\| h_\theta(x_t, t) + \frac{\tilde{\beta}_t}{\tilde{\alpha}_t} x_t - \frac{1}{\tilde{\alpha}_t} x_{t_{\min}} \right\|_2^2\right], \tag{76}$$

which expands to the loss in the proposition. ∎

Remark 1. Both losses reduce to the canonical MSE diffusion loss [54] when $t_{\min} = 0$.

Remark 2. For both the ϵ- and x_0-prediction losses, the rescaling factor, $1/\alpha_t^2$ (resp. $1/\tilde{\alpha}_t^2$), diverges as $t \to t_{\min}$. This can cause training instability. Following [19], we introduce a buffer B: a sample from $\mathcal{D}_q$ is only used at diffusion times satisfying $t > t_{\min}$ and $1/\alpha_t^2 < B$ (resp. $1/\tilde{\alpha}_t^2 < B$). The effect of B is studied in Figure 16e.

Spectral Power Law: Figures 4a and 6 visualize the spectral power law for several robot datasets. These datasets include human teleop vs algorithmically generated actions, absolute vs delta actions, end effector vs joint space actions, different parameterizations of SE(3) end-effector poses, and data from different tasks. Remarkably, a spectral power law existed in all the datasets we tested. Table I outlines the characteristics of each dataset in more detail.

The average PSD for each dataset was computed as follows. Given a dataset, we normalize the range of each action dimension to [-1,1]. For OXE, we instead normalized the 2% and 98% quartiles to [-1,1]. We do not normalize the rotation dimensions if they are represented in $\mathbb{R}^9$ format since these values already lie within [0,1] by construction. All the datasets were collected at 10 Hz; however, action frequencies in OXE vary. To ensure consistency, we set the frequency of OXE to 10 Hz by downsampling higher-frequency data and artificially speeding up lower-frequency data. We chose an action horizon of 100. For each action dimension, we compute the periodogram along the time axis. Finally, we average these periodograms across all action chunks in the dataset. Our plots exclude the DC component, which is often close to 0 due to the range normalization. These plots use a log-log scale; a power law, $f^{-\alpha}$, manifests as a straight line of slope $-\alpha$.

We fix the action frequency and horizon for consistency across datasets, but remark that the spectral power law emerges regardless of these choices. Concurrent work performed the analysis using the discrete cosine transform and obtained similar results [78].

Region Classification Experiment (from Section A): We test how well the global plan in A_0 can be recovered from A_t using a learned classifier. We train the classifier on 5000 trajectories generated by GCS trajectory optimization [65] in the 2D maze. GCS decomposes the planning problem into discrete decisions (which regions of space to move through) and continuous decisions (the precise trajectory to follow within each region). Thus, the set of regions a trajectory passes through acts as a proxy for the global planning decisions, as it captures the topological structure of the maze.

Concretely, we train a classifier $c_\theta : \mathcal{A} \times \mathbb{R} \to \mathbb{R}^{20}$ that takes as input (A_t, t) and predicts the probability that A_0 passes through each of the 20 convex regions[3]. Figure 7b shows an example trajectory and its corresponding region labels. Figure 7c plots the average prediction accuracy at each noise level on a held-out test dataset: it saturates quickly as the noise decreases. This implies that nearly all the global planning information is diffused early in the backward process. Altogether, this experiment supports our claim that action diffusion exhibits a global-to-local hierarchy.

Related Work: Decomposing action generation hierarchically is a classic approach for solving problems in robotics [79]. Recent works have begun incorporating this idea into Diffusion Policy. Generally, these approaches factorize action generation into a high and low level policy [2, 3]; however, this factorization is often handcrafted and task-specific [80, 81, 82]. Our results show that a global-to-local action hierarchy emerges naturally from the spectral structure of the data. Thus, we *hypothesize* that with sufficient data, an end-to-end policy could potentially learn the "correct" hierarchy in the diffusion process without manual design.

Locality (Sensitivity Field from Figure 4b): Let $h_\theta^{(8)}$ denote an x_0-predictor's estimate of the 8th clean action and let $A_t^{(i)}$ denote the i-th action in the noisy action chunk A_t. For each diffusion time t, the sensitivity field is defined as:

$$S_t(i) := \mathbb{E}_{(O,A_0)\sim\mathcal{D}}\left[\left\| \frac{\partial h_\theta^{(8)}(A_t, O, t)}{\partial A_t^{(i)}} \right\|_F\right], \quad i \in [0, 15]. \tag{77}$$

We normalize each sensitivity field so that $\max_i S_t(i) = 1$ at each diffusion time. Intuitively, $S_t(i)$ measures the sensitivity of the 8th action estimate to the i-th noisy action at diffusion time t.

H. Ambient Diffusion Policy Pseudocode

Given $\mathcal{D}_p$, $\mathcal{D}_q$, and annotations from Phase 1 in the Method section, we train an Ambient Diffusion Policy following Algorithm 1. At each step, we sample a batch of diffusion times $t^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}[0, T]$ and then draw admissible samples from $\mathcal{D}_p \cup \mathcal{D}_q$. We update the policy parameters, θ, to minimize the loss in COMPUTELOSS.

COMPUTELOSS executes the forward process and can use two choices for $\mathcal{L}$. The first is the standard denoising loss, $\mathcal{L}_{\text{denoising}}$ [54]. The second is the *Ambient loss*, $\mathcal{L}_{\text{ambient}}$, discussed in Appendix G0a.[4] We present two versions of COMPUTELOSS for variance-exploding (Algorithm 2) and variance-preserving diffusion processes (Algorithm 3).

I. Diffusion Policy Implementation Details

Unless otherwise noted, policies are trained to predict the next 16 actions at 10 Hz with a fixed horizon for observations and actions. We execute the first 8 actions and predict new

[3]We intentionally exclude the start and goal conditioning from the classifier since those are sufficient for predicting the global plan.

[4]The Ambient loss was only used in the maze experiments (see Table II). We tested the Ambient loss on the experiments in the OXE experiments and found it made no statistically significant difference; thus, we continued using the regular denoising loss for simplicity.

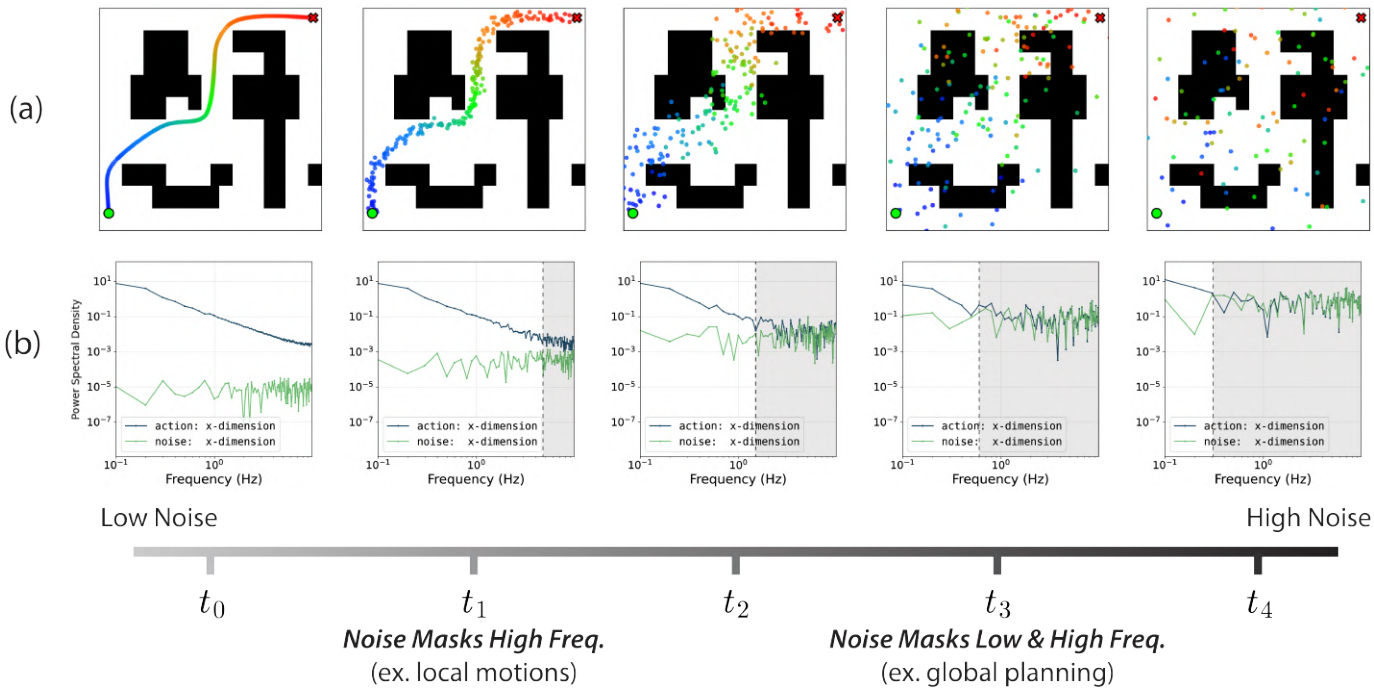
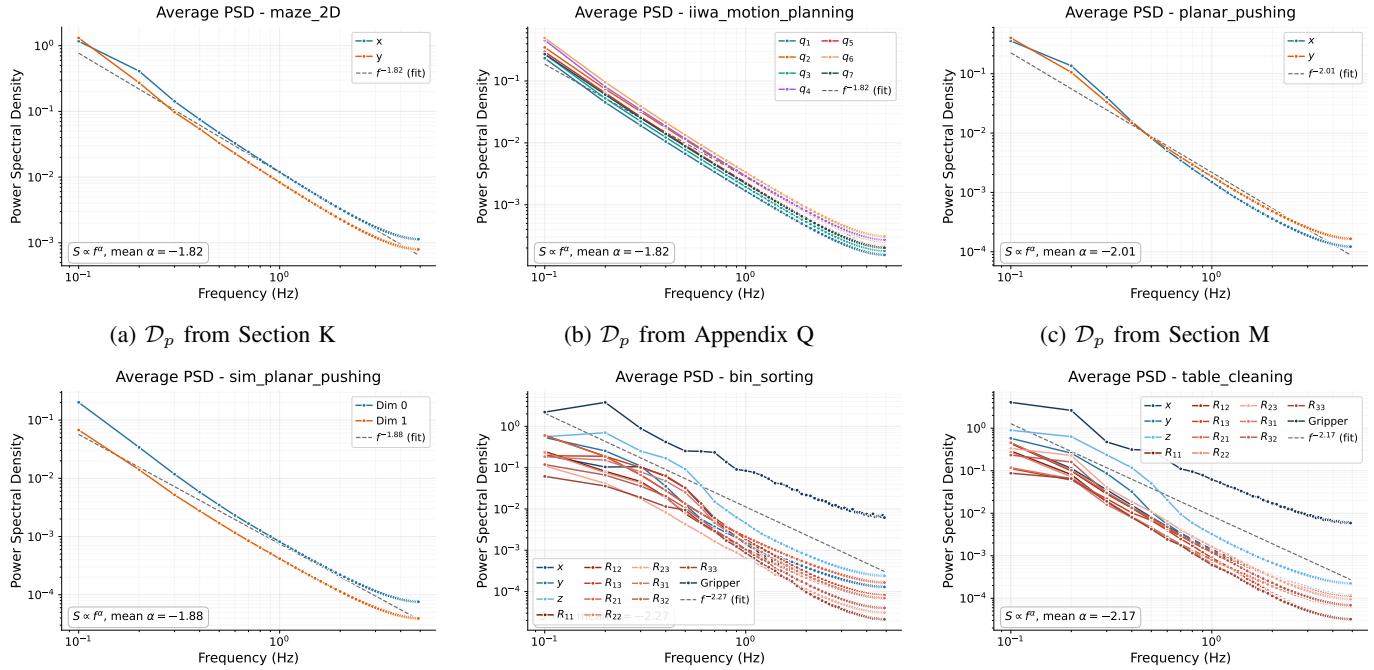


Fig. 5: **The spectral power law makes Gaussian noise act as a high-frequency mask. (a)** Noisy trajectory visualizations. **(b)** PSD analysis for the noisy trajectories in the x-dimension at each diffusion time: the gray regions indicate *masked* frequencies, which have low SNR. As t increases, adding Gaussian noise destroys the signal in the trajectory, starting from the local (high-frequency) features first. This effect is visible in both the noisy trajectories and the PSD plots.



(a) $\mathcal{D}_p$ from Section K (b) $\mathcal{D}_p$ from Appendix Q (c) $\mathcal{D}_p$ from Section M

(d) $\mathcal{D}_q$ from Section M (e) $\mathcal{D}_p$ from Section N (f) $\mathcal{D}_p$ from Section V

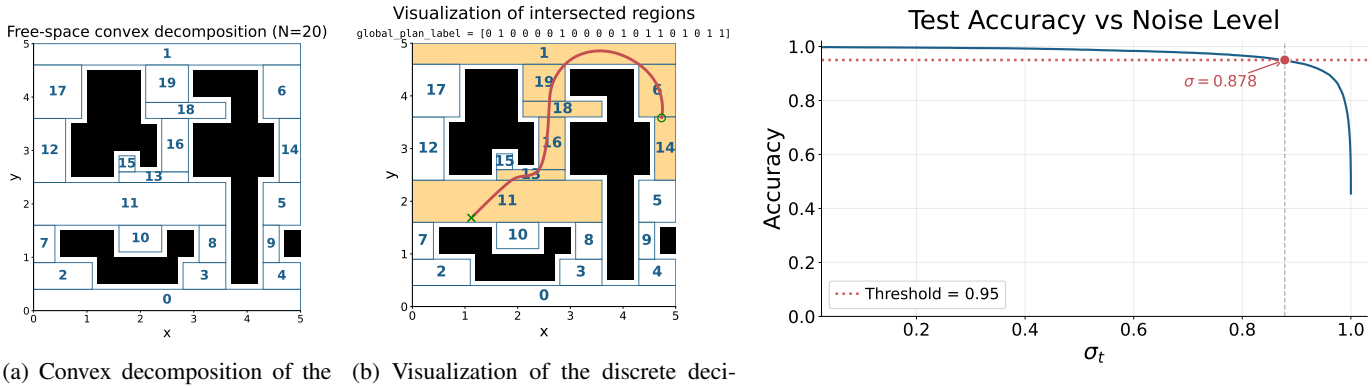
Fig. 6: **A visualization of the average PSD for a diverse range of robot datasets; a spectral power law exists in all of them.** Table I characterizes the datasets shown in (a)-(f) as well as the OXE dataset shown in Figure 4a.

actions in an MPC-like fashion. Policies are conditioned on the 2 most recent observations. We consider a fixed horizon for observations and actions and assume that the learned policy is *time-invariant*, i.e. the conditional action distribution $\pi(\cdot \mid O)$

TABLE I: Characteristics of the datasets analyzed via power spectral density (PSD) in Figures 4a and 6. Despite their differences, all datasets exhibit a spectral power law.

Fig.	Dataset	Source	Generation	Action Space	Action Type	Rotation Rep.	$\bar{\alpha}^\dagger$
Fig. 4a	COXE [1]	$\mathcal{D}_q$ (Sec. V)	Mixed	Task space	Delta	Axis-angle	-1.47
Fig. 6a	GCS [65]	$\mathcal{D}_p$ (Sec. K)	Algorithmic	x-y	Absolute	N/A	-1.82
Fig. 6b	TrajOpt.	$\mathcal{D}_p$ (App. Q)	Algorithmic	Joint space	Absolute	N/A	-1.82
Fig. 6c	Planar pushing [12]	$\mathcal{D}_p$ (Sec. M)	Human teleop	x-y	Absolute	N/A	-2.01
Fig. 6d	Graesdal et al. [72]	$\mathcal{D}_q$ (Sec. M)	Algorithmic	x-y	Absolute	N/A	-1.88
Fig. 6e	Block sorting	$\mathcal{D}_p$ (Sec. N)	Human teleop	Task space	Absolute	$\mathbb{R}^9$	-2.27
Fig. 6f	Table cleaning	$\mathcal{D}_p$ (Sec. V)	Human teleop	Task space	Absolute	$\mathbb{R}^9$	-2.17

† $\bar{\alpha}$: mean fitted exponent across action dimensions, where $S(f) \propto f^{\alpha}$



(a) Convex decomposition of the collision-free space.

(b) Visualization of the discrete decisions in GCS.

(c) Prediction accuracy for the plan at different noise levels.

Fig. 7: **The backward process commits to a global plan at high noise.** By $\sigma_t = 0.878$, the global plan in A_0 can already be recovered from A_t with 95% accuracy. Accuracy plateaus afterward. This indicates that the start of the backward process performs global planning. Figure 3 shows that the remainder of the backward process is devoted to local trajectory refinement.

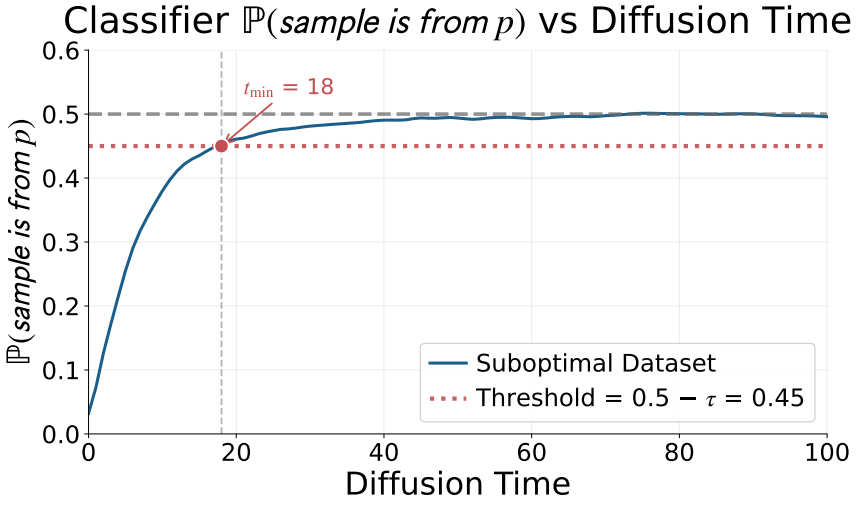


Fig. 8: **Average classifier output for suboptimal samples in the 2D maze dataset (Section K) at different diffusion times.** As t increases, p_t and q_t become harder to distinguish. The dotted line is the threshold $0.5 - \tau$, which is crossed for $t_{\min} = 18$.

depends only on the most recent observations and not on the time t. The per-experiment observation and action spaces are summarized in Tables II and III.

We use the U-Net architecture from [23], but add the option for conditioning on a goal. When a policy is image-conditioned, we use the ResNet18 backbone for the vision encoder. We train the policy end-to-end with a cosine learning rate schedule. The noise schedule $\sigma(t)$ is a cosine schedule with $T = 100$ diffusion steps. We apply EMA to the training loop. Sampling is performed online using 10 steps of DDIM [57]. Predicted actions are linearly interpolated into piecewise linear trajectories and passed through a first-order low-pass filter before being sent to the robot.

Algorithm 1 Ambient Diffusion Policy: Training

Require: $\mathcal{D}_p$, $\mathcal{D}_q$ with $(t^{(i)}_{\min}, t^{(i)}_{\max})$, $\mathcal{L} \in \{\mathcal{L}_{\text{ambient}}, \mathcal{L}_{\text{denoising}}\}$, noise schedule $\sigma(t)$, max time T, batch size B, policy parameters θ

repeat
  Sample $t^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}[0,T]$ for $i = 1, \dots, B$
  Sample batch $\{(O^{(i)}, A_0^{(i)})\}_{i=1}^{B}$ s.t. $t^{(i)} \in [0, t^{(i)}_{\max}) \cup (t^{(i)}_{\min}, T]$
  $\ell \leftarrow \frac{1}{B}\sum_i \textsc{ComputeLoss}(\mathcal{L}, A_0^{(i)}, O^{(i)}, t^{(i)}, t^{(i)}_{\max}, t^{(i)}_{\min})$
  Update θ on ℓ
until converged
return π_θ

Algorithm 2 ComputeLoss (variance-exploding)

Require: $\mathcal{L}$, A_0, O, t, $t_{\max}$, $t_{\min}$

if $t \in [0, t_{\max})$ **or** $\mathcal{L} = \mathcal{L}_{\text{denoising}}$ **then**
  $A_t \leftarrow A_0 + \sigma_t Z, \quad Z \sim \mathcal{N}(0, I)$
  return $\mathcal{L}_{\text{denoising}}(A_t, A_0, O, t)$ ▷ standard DDPM loss [54]
else ▷ $t \in (t_{\min}, T]$ and $\mathcal{L} = \mathcal{L}^{\text{VE}}_{\text{ambient}}$
  Sample $Z_1, Z_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$
  $A_{t_{\min}} \leftarrow A_0 + \sigma_{t_{\min}} Z_1$
  $A_t \leftarrow A_{t_{\min}} + \sqrt{\sigma_t^2 - \sigma_{t_{\min}}^2}\, Z_2$
  return $\mathcal{L}^{\text{VE}}_{\text{ambient}}(A_t, A_{t_{\min}}, O, t)$ ▷ VE Ambient loss [24]
end if

J. Future Direction: Classifier-Based Rejection Sampling

This section proposes a similar algorithm to Ambient Diffusion Policy based on rejection sampling. We provide theoretical foundations for this method, but do not experimentally verify it. This is an exciting direction for future work.

Recall that we trained a classifier that outputs the probability that a noisy action chunk came from p_t (as opposed to q_t). Algorithm 4 samples (approximately) from P given only access to samples from the mixture $M = \frac{1}{2}P + \frac{1}{2}Q$ and the classifier. Proposition 6 bounds the total variation between the sampled distribution and P. By using the classifier for rejection sampling, we can train denoisers for P despite leveraging data from an auxiliary source Q. In the proofs, we use f_θ to denote the classifier instead of c_ϕ from Section IV.

Algorithm 3 ComputeLoss (variance-preserving)

Require: $\mathcal{L}$, A_0, O, t, $t_{\max}$, $t_{\min}$

if $t \in [0, t_{\max})$ **or** $\mathcal{L} = \mathcal{L}_{\text{denoising}}$ **then**
  $A_t \leftarrow \sqrt{1-\sigma_t^2}A_0 + \sigma_t Z, \quad Z \sim \mathcal{N}(0, I)$
  return $\mathcal{L}_{\text{denoising}}(A_t, A_0, O, t)$ ▷ standard DDPM loss [54]
else ▷ $t \in (t_{\min}, T]$ and $\mathcal{L} = \mathcal{L}^{\text{VP}}_{\text{ambient}}$
  Sample $Z_1, Z_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$
  $A_{t_{\min}} \leftarrow \sqrt{1-\sigma_{t_{\min}}^2}A_0 + \sigma_{t_{\min}} Z_1$
  $A_t \leftarrow \sqrt{\frac{1-\sigma_t^2}{1-\sigma_{t_{\min}}^2}}A_{t_{\min}} + \sqrt{\frac{\sigma_t^2 - \sigma_{t_{\min}}^2}{1-\sigma_{t_{\min}}^2}}Z_2$
  return $\mathcal{L}^{\text{VP}}_{\text{ambient}}(A_t, A_{t_{\min}}, O, t)$ ▷ Eq. (64) (ϵ-pred) or Eq. (72) (x_0-pred)
end if

Algorithm 4 Rejection sampling via classifier

Require: Noisy classifier $f_\theta(x, \sigma)$, mixture distribution $M = \frac{1}{2}P + \frac{1}{2}Q$, noise distribution ρ_σ

Sample $\sigma \sim \rho_\sigma$
while True **do**
  Sample $X \sim M$
  $X_\sigma \leftarrow \textsc{AddNoise}(X, \sigma)$
  Sample $c \sim \text{Bern}(f_\theta(X_\sigma, \sigma))$
  if $c = 1$ **then**
    return (σ, X_σ)
  end if
end while

Proposition 6 (Rejection sampling guarantee)**.** *Suppose f_θ is uniformly ε-optimal across σ:*

$$\mathcal{L}_{\text{CE}}(f \mid \sigma) \le \mathcal{L}_{\text{CE}}(f^* \mid \sigma) + \varepsilon, \qquad \forall \sigma.$$

For any $\sigma \ge 0$, let P_σ be the noisy distribution of $X \sim P$, and let $\hat{P}_\sigma$ be the distribution of X_σ output by Algorithm 4. Then

$$\mathrm{d}_{\text{TV}}(\hat{P}_\sigma, P_\sigma) \le O(\sqrt{\varepsilon}).$$

Proof: We drop the subscript σ and write $P \equiv P_\sigma$, $Q \equiv Q_\sigma$. Let $m(x)$ denote the mixture density. The true distribution P satisfies $p(x) = 2m(x)f^*(x)$, while the rejection-sampled density is

$$\hat{p}(x) = \frac{m(x)f_\theta(x)}{Z}, \quad Z = \mathbb{E}_{X\sim M}[f_\theta(X)],$$

where Z is the acceptance rate. By Pinsker applied to the cross-entropy gap, $\mathbb{E}_{X\sim M}[(f^*(X) - f_\theta(X))^2] \le \varepsilon/2$, so by Jensen

$$\delta \triangleq \mathbb{E}_{X\sim M}|f_\theta(X) - f^*(X)| \le \sqrt{\varepsilon/2}.$$

This gives $|Z - \frac{1}{2}| \le \delta$, hence $Z \ge \frac{1}{2} - \delta$. The total variation between $\hat{P}$ and P is

$$\begin{aligned}\mathrm{d}_{\text{TV}}(\hat{P}, P) &= \frac{1}{2}\int \left|\frac{m(x)f_\theta(x)}{Z} - 2m(x)f^*(x)\right| dx \\ &= \mathbb{E}_{X\sim M}\left|\frac{f_\theta(X) - 2Zf^*(X)}{2Z}\right|.\end{aligned}$$

By the triangle inequality,

$$\left|\frac{f_\theta(X) - 2Zf^*(X)}{2Z}\right| \le \frac{|f_\theta(X) - f^*(X)|}{2Z} + \frac{f^*(X)}{2Z}|1 - 2Z|.$$

Taking the expectation over M and using $\mathbb{E}_{X\sim M}[f^*(X)] = \frac{1}{2}$ and $|1 - 2Z| \le 2\delta$:

$$\mathrm{d}_{\text{TV}}(\hat{P}, P) \le \frac{\delta}{2Z} + \frac{2\delta}{4Z} = \frac{\delta}{Z} \le \frac{2\delta}{1 - 2\delta}.$$

For $\varepsilon < 2/9$ we have $1 - 2\delta > 1/3$, hence

$$\mathrm{d_{TV}}(\hat{P}, P) \leq 6\sqrt{\varepsilon/2} = O(\sqrt{\varepsilon}).$$

■

Tables II and III summarize the datasets, distribution shifts, observation and action spaces, hyperparameters, loss functions, and evaluation details for all main experiments in Sections J and V respectively; ablation parameters may differ. Importantly, we swept hyperparameters for each baseline to ensure a fair comparison.

Binary success metrics are reported with 95% Clopper-Pearson confidence intervals. Otherwise, error bars are standard error intervals. For each training run, we save 5-10 checkpoints and report the performance of the best-performing one. While the initial exposition of our algorithm was presented with variance-exploding diffusion, our implementation uses variance-preserving diffusion (i.e. $\sigma(T) = 1$).

We evaluate Ambient Diffusion Policy on three distribution shifts in robot action data to illustrate its generality: noisy trajectories, sim-to-real gap, and task mismatch. This section compares our method against data filtering, co-training, and finetuning. Appendix J provides experimental details.

K. Noisy Trajectories

Consider a 2D point robot that must navigate between random start and goal positions in a maze environment (Figure 1a). A policy rollout is successful if it reaches an ϵ-ball around the goal within a time limit without collision. We measure a policy's *global* understanding of the maze using success rate, and its *local* smoothness using average squared acceleration (Eq. (78)).

$\mathcal{D}_p$ contains 50 smooth trajectories from GCS [65]; $\mathcal{D}_q$ contains 5,000 cheaper but jittery trajectories from RRT [66]. Appendix Figure 9 visualizes the datasets. This mimics real-world sources of suboptimality, such as teleoperation with low-quality hardware or noisy control stacks [67].

The policy trained with data filtering is smooth, but performs poorly (Table IVa). Co-training and Ambient both perform well ($99.0\%+$ success), but the Ambient Diffusion Policy is $2\times$ smoother. This is because RRT's jittery behavior is a local property; thus, excluding it from training at low diffusion times prevents the policy from learning its non-smooth behavior. Figure 11 (Appendix P) visualizes the rollouts. Lastly, training on $\mathcal{D}_q$ trades smoothness for better data coverage and success rates. Ambient's Pareto frontier for this trade-off strictly dominates the co-training baseline (Appendix Figure 10a).

Ablation: We claim that $p_t \approx q_t$ for $t > t_{\min}$. If this were true, then we should be able to train an effective policy without using $\mathcal{D}_p$ for $t > t_{\min}$. To test this, we train a policy that exclusively trains on $\mathcal{D}_q$ for $t \in (t_{\min}, T]$ and $\mathcal{D}_p$ for $t \in [0, t_{\min}]$. Remarkably, it matches the performance of the Ambient Diffusion Policy, with a success rate of 99.0% and a squared acceleration of 29.9.

L. Neural Motion Planning

We extend the maze experiment to neural motion planning [68, 69, 70] with a 7-DoF robot arm in a cluttered environment (Figure 1b). The setup and metrics are analogous to the 2D maze, except that start and goal configurations are sampled around objects and shelves, and the planner predicts the entire plan open-loop. Appendix Table VII compares the two setups. The main results in Table IVb and Appendix Figure 10b mirror the 2D maze experiments: Ambient achieves the highest success rate while remaining substantially smoother than co-training. The full experimental details and ablations are in Appendix Q.

Implications for Neural Motion Planning (NMP). Most neural motion planners [71, 69] are trained on large datasets of sampling-based trajectories (akin to $\mathcal{D}_q$) since they are inexpensive to generate. Unfortunately, sampling-based trajectories are non-smooth [66]; thus, these approaches choose to prioritize data scale over quality. Our results suggest that an NMP trained with Ambient can enjoy the best of both worlds: augmenting the existing sampling-based datasets with a small amount of high-quality data could yield planners that are both general and smooth.

M. Sim-to-Real Distribution Shift

We evaluate Ambient Diffusion Policy on sim-and-real co-training using the planar pushing setup from [12] (Figure 1c), a canonical task in robot imitation learning. Our experiments use the *sim-and-target* problem from Wei et al. [12] as a controlled proxy for the sim-and-real problem.

$\mathcal{D}_p$ consists of 50 teleoperated demonstrations in the *target* environment; $\mathcal{D}_q$ consists of 2000 trajectories generated by a motion planner [72] in the *simulated* environment. We use the same datasets from Wei et al. [12] to enable direct comparisons with their method. A policy rollout is successful if it pushes the T to the target pose within a time limit.

Table V reports the success rate for three variations of Ambient Diffusion Policy that annotate $t_{\min}$ at two granularities: *per dataset* (with a parameter sweep) and *per datapoint* (with the classifier approach). We explore intermediate granularities, scaling laws, and more ablations in Appendix R. All Ambient variations outperform the best co-trained policy [12]. Annotating $t_{\min}$ per datapoint performs the best because even within a single dataset, the utility of each datapoint can vary. As before, Ambient learns from suboptimal simulation data by leveraging the diffusion hierarchy. $\mathcal{D}_p$ and $\mathcal{D}_q$ share the same high-level strategy; thus, $\mathcal{D}_q$ is safe to use at high noise. Locality ($t_{\max} > 0$) does not help since $\mathcal{D}_p$ and $\mathcal{D}_q$ exhibit different low-level contact dynamics.

N. Task Mismatch

Ambient can leverage "locality" to learn local action primitives (e.g., grasping) from data collected on a different task. Consider the block sorting task in Figure 1d. The robot must sort red and blue blocks into the left and right bins, respectively. This requires two distinct skills: *motion* (grasping

and placing blocks) and *logic* (selecting the correct bin). We design a metric to evaluate each skill:

- **Motion metric:** $\frac{\text{\# blocks placed}}{\text{\# total blocks}}$. This metric is independent of logical reasoning because the numerator counts any placed block, regardless of logical correctness.
- **Logic metric:** $\frac{\text{\# blocks placed correctly}}{\text{\# blocks placed}}$. This metric is independent of pick-and-place ability because the denominator only counts successfully placed blocks.

The overall success rate is $\frac{\text{\# blocks placed correctly}}{\text{\# total blocks}}$, the product of the two metrics. $\mathcal{D}_p$ contains 50 demonstrations with the correct sorting. $\mathcal{D}_q$ contains 200 demonstrations with the opposite sorting. To isolate the effect of locality, we set $t_{\min} = 0$ and sweep $t_{\max}$ for $\mathcal{D}_q$.

Table VI presents the results. Co-training performs poorly because training on $\mathcal{D}_q$ at all diffusion times teaches the policy both the useful motion primitives and the incorrect logic. Ambient Diffusion Policy solves this problem by restricting $\mathcal{D}_q$ to $t \in [0, t_{\max})$. As illustrated in Figure 3 and Appendix Figure 17, the policy learns motion primitives at low diffusion times without attending to the (incorrect) task-level structure of the demonstrations. This results in near-perfect scores on both metrics simultaneously.

Ablations: We present two ablations in Appendix S. **1)** we add a one-hot encoding indicating the sorting direction to the policies. Ambient still outperforms the co-training. **2)** we train a policy that exclusively uses $\mathcal{D}_q$ for $t \in [0, t_{\max})$ and $\mathcal{D}_p$ for $t \in [t_{\max}, T]$. This is the *locality*-version of the ablation in the 2D maze experiment. The resulting policy achieves a logic and motion score of 97.9% and 93.8%, reinforcing our claim that action diffusion is hierarchical.

O. Finetuning Comparison

Finally, we compare against the finetuning baseline. For each experiment, we finetune the best co-trained policy and the best Ambient policy on $\mathcal{D}_p$; implementation details are in Appendix V.

Appendix Figure 22 suggests three findings. First, finetuning an Ambient base policy consistently outperforms finetuning a co-trained base policy. Second, the Ambient *base* policy can often outperform the *finetuned* co-trained policy. Third, finetuning an Ambient base policy does not always help. We hypothesize that the Ambient policies already ignore undesirable features in the suboptimal data.

P. Noisy Trajectories: 2D Maze

Data Generation: The GCS [65] trajectories were generated using third-order Bezier curves, first-order continuity constraints, and a maximum of 10 rounded paths. Both the GCS and the RRT datasets were generated with a velocity constraint of 1m/s in the x and y directions and 0.1m of obstacle padding. The maze environment is 5m by 5m.

Evaluation: During each trial, we sample the start and goal uniformly in the padded collision-free configuration space; the robot is initialized at the start configuration. A trial is successful if the robot can navigate to a 0.15m ball around the end configuration without collision within 30s of simulation time. We performed 1000 trials per policy.

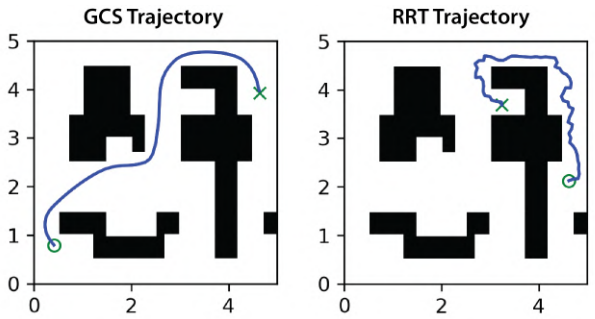


Fig. 9: **GCS (left) vs RRT (right) data.** The high-quality GCS [65] trajectories are smoother and shorter than their RRT [66] counterparts.

We compute the smoothness metric as the average squared acceleration of the policy's rollout. Concretely, we fit a cubic spline to the rollout and evaluate

$$\text{Smoothness} = \tfrac{1}{T}\int_0^T \|a(t)\|_2^2 \, dt, \tag{78}$$

where $a(t)$ is the acceleration of the spline at time t. We only report smoothness for the successful rollouts, for three reasons: 1) failed rollouts have orders of magnitude higher accelerations that disproportionally bias the results, 2) across 1000 trials, most policies had less than 10 failed rollouts, 3) the overall trend when averaging across all trials is the same.

Figure 12 shows the maze success rate as a function of $\sigma_{t_{\min}}$, and Figure 11 visualizes representative policy rollouts for each training algorithm. The smoothness–success-rate Pareto frontiers for the maze and neural motion planning experiments are shown in Figure 10.

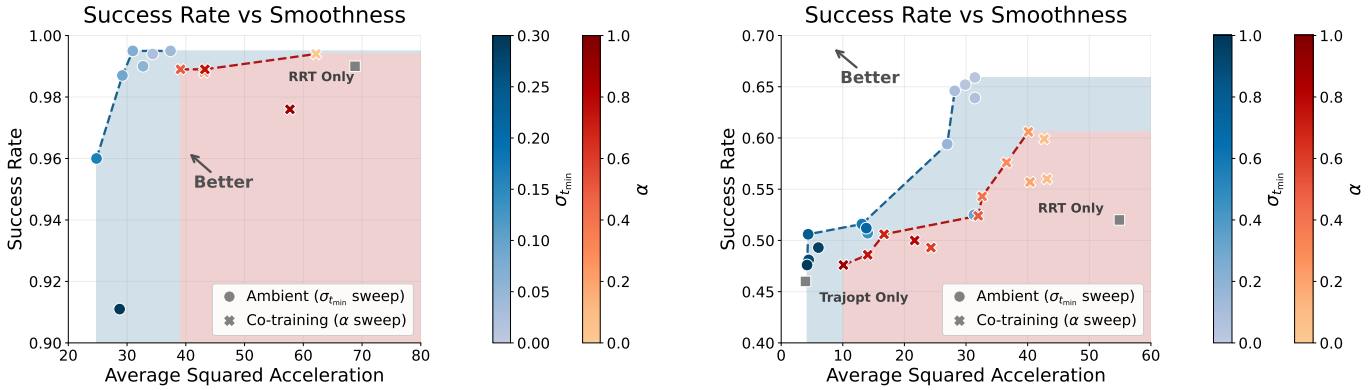
Q. Noisy Trajectories: Neural Motion Planning

This appendix provides the full experimental details for the neural motion planning experiment in Section L, along with best-of-100 results and a $\sigma_{t_{\min}}$ sweep. The main results (Table IVb and Figure 10b) are presented in Section L. Table VII compares the 2D maze and 7-DoF NMP setups side by side.

The environment in Figure 1b was generated using a procedural method [85] and the assets are from Pfaff et al. [86]. Start and goal configurations are determined by sampling a random end-effector target point $p \in \mathrm{SE}(3)$ and then solving the optimization inverse kinematics problem

$$\begin{aligned} \text{find} \quad & q \in \mathbb{R}^7 \\ \text{s.t.} \quad & \mathrm{FK}_{\mathrm{pos}}(q) = p, \\ & \mathrm{SDF}(q) \geq 0.01, \end{aligned} \tag{79}$$

where $\mathrm{FK}_{\mathrm{pos}}$ is the position forward kinematics, returning the position of the gripper as a function of the joint angles, and SDF returns the clearance (in meters) between the robot and any obstacles. The constraint $\mathrm{SDF}(q) \geq 0.01$ is implemented via Drake's



(a) 2D maze.

(b) 7-DoF neural motion planning.

Fig. 10: **The "smoothness vs success rate" Pareto frontier for Ambient (blue) dominates co-training (red)** in both the (a) maze and (b) neural motion planning experiments. This trade-off is controlled by $\sigma_{t_{\min}}$ for Ambient and α for co-training.

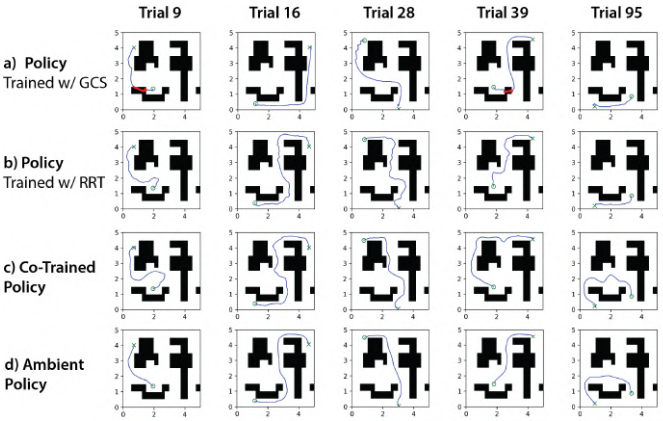


Fig. 11: **Policy rollouts from the same start and goal pairs for different training algorithms. a)** The policy trained with data filtering (i.e. GCS-only) has not seen enough data to learn the maze. Its rollouts frequently collide with obstacles. **b)** The policy trained with RRT-only has memorized the maze, but exhibits non-smoothness. **c)** The co-trained policy's behavior is similar to the RRT-only policy. **d)** The Ambient Diffusion Policy is smooth and understands the structure of the maze.

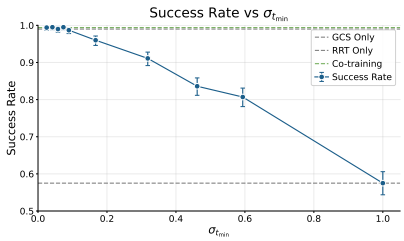


Fig. 12: **Maze success rate vs** $\sigma_{t_{\min}}$. For small $\sigma_{t_{\min}}$, the Ambient Policies are smooth and achieve high success rate (see Figure 10a). As $\sigma_{t_{\min}}$ increases, the RRT data is used less during training, which reduces success rate.

`MinimumDistanceLowerBoundConstraint`. Targets are sampled 1) within shelves, 2) in bins, 3) near objects, and 4) uniformly in the free space. We sample from all 4 classes of target configurations with equal probability. (79) is solved using SNOPT [87].

Given a start and goal configuration $q_0, q_1 \in \mathbb{R}^7$, we use BiRRT [88] to produce the suboptimal data. To produce the high-quality data, we first run randomized shortcutting [89] to slightly simplify the path. We then solve the trajectory optimization problem

$$\begin{aligned} \min_{q(s)} \quad & \int_0^1 ||\dot{q}(s)||^2 \, \mathrm{d}s \\ \text{s.t.} \quad & q(0) = q_0, q(1) = q_1, \\ & \mathrm{SDF}(q(s)) \geq 0, \ \forall s \in [0,1] \end{aligned} \tag{80}$$

with Drake's `KinematicTrajectoryOptimization`, using its standard B-spline discretization [90]. We use the shortcut trajectory as an initial guess and solve with IPOPT [91], with solver options tuned aggressively to maintain feasibility.

The collision-free constraint $\mathrm{SDF}(q(s)) \geq 0, \ \forall s \in [0,1]$ is applied at 200 points evenly spaced along the trajectory. Collision checking with finer granularity is performed after trajectory optimization to filter trajectories with collisions. In principle, the discarded trajectories could be added to $\mathcal{D}_q$ and used at sufficiently high diffusion times during training.

Each trajectory in $\mathcal{D}_p$ and $\mathcal{D}_q$ is subsampled to 100 waypoints, equally spaced in time. This subsampling procedure acts as a low-pass filter which reduces some of the non-smoothness in the BiRRT data.

Evaluation: Instead of rolling out the model closed-loop (as in the maze experiments), we train the planner to predict the entire plan at once. Start and end configurations are sampled from the 4 classes described above. Each trial is successful if the planner predicts a collision-free trajectory with up to 2cm of penetration. We allow 2cm of padding because the training trajectories were generated with 0cm of padding and frequently pass very close to obstacles. Both the first and last waypoints of the predicted plan must be within a 0.2rad ball of the start and end configurations respectively.

Training Algo.	Success Rate ↑	Smoothness ↓ (Eq. (78))
Data Filtering	$73.9^{+2.7}_{-2.8}$ %	**3.7**
Co-train ($\alpha^* = 0.091$)	$91.3^{+1.7}_{-1.9}$ %	32.6
Ambient ($\sigma_{t^*_{\min}} = 0.025$)	$\mathbf{91.9^{+1.6}_{-1.9}}$ %	20.2

TABLE VIII: **Neural Motion Planning Best-of-100 Results (1000 trials).** Ambient retains the highest success rate while achieving lower squared acceleration than the co-trained baseline.

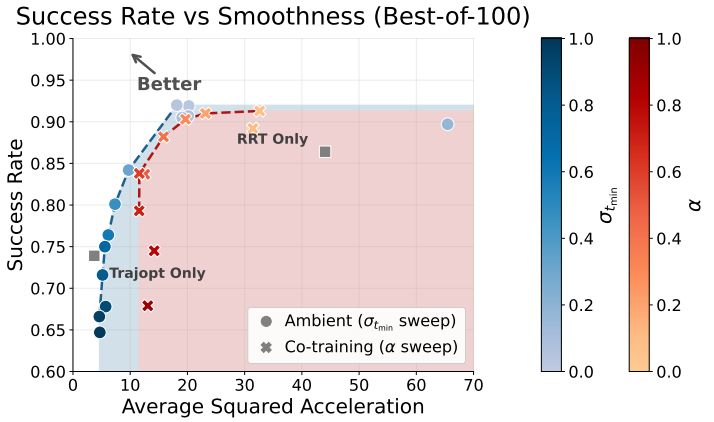
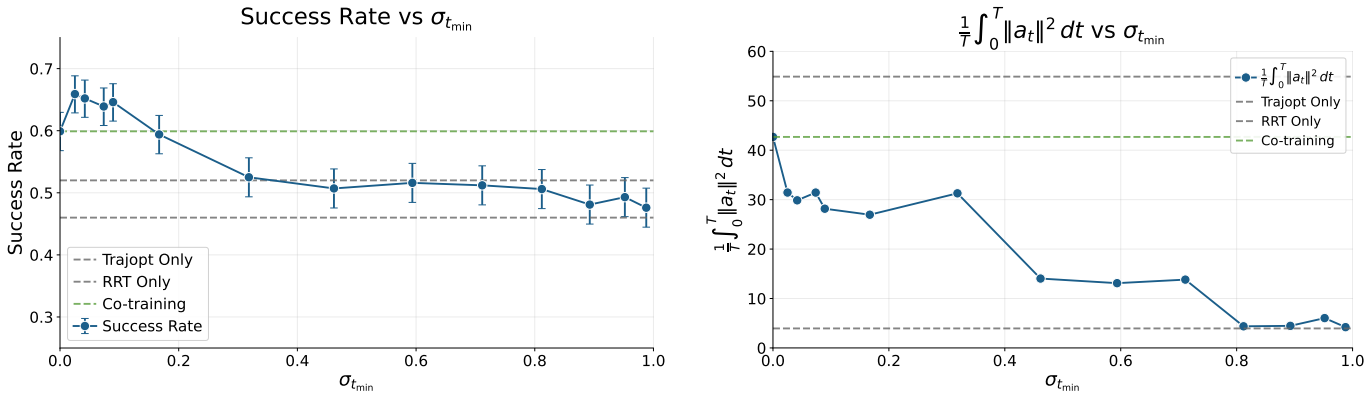


Fig. 13: **Figure 10b repeated with best-of-100 evaluation.** The overall trends are the same, but the success rates are higher and the gap between the two Pareto frontiers is smaller.



(a) Success rate vs $\sigma_{t_{\min}}$.

(b) Smoothness metric vs $\sigma_{t_{\min}}$.

Fig. 14: **Neural motion planning $\sigma_{t_{\min}}$ sweep. (a)** For small $\sigma_{t_{\min}}$, the policies are smoother, which results in fewer accidental collisions and increases success rate. For large $\sigma_{t_{\min}}$, the RRT data becomes under-utilized, which decreases the success rate. **(b)** Increasing $\sigma_{t_{\min}}$ improves smoothness.

Previous works in neural motion planning adopt a best-of-N approach when evaluating their planners [71]. We present the results using $N = 1$ (Table IVb and Figure 10b) and $N = 100$ as in [68] (Table VIII and Figure 13). Concretely, for each trial, we generate 100 plans and pick the one with the least collisions. The general trends for both choices of N are the same, but the success rates are significantly higher. The Ambient policy continues to be smoother than the co-trained policy in both cases.

R. Sim-and-Real Co-training: Planar Pushing

Data Generation: We use the datasets from Wei et al. [12]. $\mathcal{D}_p$ contains human teleoperated demonstrations in their "target" simulation environment and $\mathcal{D}_q$ contains automatically generated demonstrations [72] from their "source" simulation environment.

Table IX shows the distribution shift between the source and target simulation datasets. For more details and visualizations of the environments, see Wei et al. [12].

Evaluation: The slider is randomly initialized within the robot's workspace. A trial is successful if the robot returns to its default position and the slider's pose is within 1.5cm and 3.5^o of the goal. The time limit is 75s. The evaluation criteria exactly match Wei et al. [12] to ensure a fair comparison.

Ablations: We use these experiments as a test-bed for evaluating key design decisions in Ambient Diffusion Policy. Figure 16 consolidates eight ablations; the results suggest the following:

- **Per-datapoint $t_{\min}$ (Figure 16a):** Labeling $\sigma_{t_{\min}}$ per dataset improves over co-training, and labeling $\sigma_{t_{\min}}$ per datapoint performs the best. The distribution of $t_{\min}$ assigned by the best classifier is shown in Figure 15.
- **$\sigma_{t_{\max}}$ sweep (Figure 16b):** Using locality ($\sigma_{t_{\max}} > 0$) does not appear to improve or hurt performance in sim-and-real co-training until $\sigma_{t_{\max}}$ is larger. Sim and real data do not share the same local action structure (e.g. due to differing contact dynamics); thus, policies should not train on sim data at low diffusion times.

- **Classifier robustness (Figures 16c and 16d):** Policy performance is insensitive to the classifier checkpoint and threshold τ; the classifier merely needs to be sufficiently trained.
- **Ambient loss buffer (Figure 16e):** The Ambient loss is highly sensitive to the buffer size from Appendix G0a. In most experiments (Table II), we avoid this sensitivity by training with the regular denoising objective [54].
- $t_{\min}$ **granularity (Figure 16f):** Assigning $t_{\min}$ at intermediate granularities (one $t_{\min}$ per partition of $\mathcal{D}_q$) appears to be a promising direction. We split $\mathcal{D}_q$ into N partitions and annotate one $\sigma_{t_{\min}}$ per partition. $N = 1$ corresponds to assigning a single $t_{\min}$ to all of $\mathcal{D}_q$, and $N \to \infty$ corresponds to annotating a $t_{\min}$ per sample. Given a partition P_i, we annotate

$$t_{\min} = \inf\{t : \mathbb{E}_{(O,A_0)\sim P_i, A_t|A_0}[c_{\phi^*}(A_t, t)] > 0.5 - \tau\}.$$

  We compare three partitioning strategies: (1) randomly partition the action chunks (blue); (2) randomly partition the trajectories – each partition contains the action chunks of its trajectories (red); (3) find $t_{\min}$ for each action chunk using the classifier and sort the chunks according to these values. Then form the partitions in increasing order (orange). For this task, Strategy (1) with $N = 10{,}000$ performed best.
- $|\mathcal{D}_q|$ **scaling (Figures 16g and 16h):** As $\mathcal{D}_q$ scales, Ambient consistently outperforms co-training via re-weighting at both $|\mathcal{D}_p| = 10$ and $|\mathcal{D}_p| = 50$ (except at $|\mathcal{D}_q| = 4000$ in the latter). The co-training baseline results (green) are from [12].

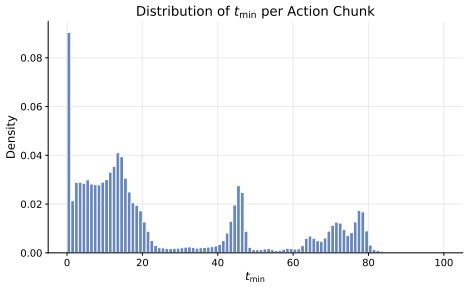


Fig. 15: **Distribution of $t_{\min}$ labels.** This histogram shows the distribution of $t_{\min} \in [0, 100]$ assigned to each action chunk by the classifier for the sim-and-real experiments.

S. Task Mismatch: Block Sorting

Data Collection: Both datasets were collected using VR teleoperation.

Evaluation: On each trial, 1-3 red blocks and 1-3 blue blocks are randomly initialized in a pile between the bins. The time limit is 20s per block. This allows trials with more blocks to have a longer time limit. Each policy is evaluated on 200 trials ($\sim$800 blocks).

Figure 17 shows the performance of the Ambient Diffusion Policy on each metric during the $\sigma_{t_{\max}}$ sweep. Figures 18a and 18b show the average number of grasp attempts and time required per block of the policies trained with different $\sigma_{t_{\max}}$ values. Both can be viewed as additional *motion* metrics and improve (decrease) with higher $\sigma_{t_{\max}}$.

Ablation 1 (Task-Conditioned): A standard solution to handling the incorrect data is to add a one-hot encoding that indicates the sorting direction. This is analogous to task or quality conditioning [10]. Even with task conditioning, the Ambient Diffusion Policy performs the best (Table Xa). Note that Ambient Diffusion Policy and task-conditioning are complementary.

Ablation 2 ($\mathcal{D}_q$ only at low noise): Locality implies that the denoisers at $t < t_{\max}$ are insensitive to the global task structure. If so, we should be able to train an effective policy without using any "correct" data at low noise. We train a policy (without task-conditioning) that exclusively uses $\mathcal{D}_q$ for $t \in [0, t_{\max})$ and $\mathcal{D}_p$ for $t \in (t_{\max}, T]$. This policy still achieves a logic accuracy of 97.9% and a motion score of 93.8% (Table Xb). This reinforces our claim that the backward process factorizes into a planning and a local refinement regime.

T. OXE Experiments

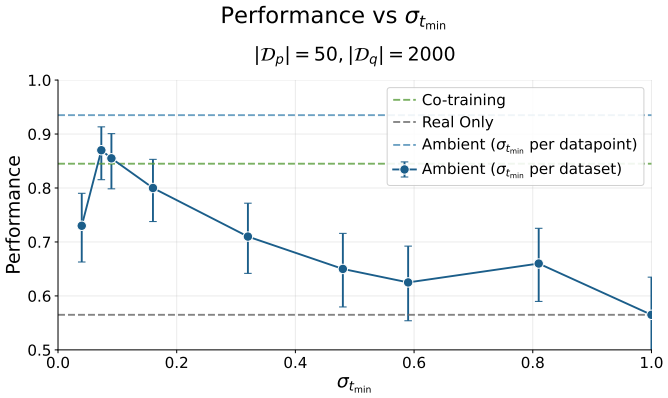
Dataset Curation: We collected demonstrations for $\mathcal{D}_p$ with VR teleoperation. For each demonstration, 3-5 objects are placed on the table. These objects are selected among a set of 31 objects in the training distribution. The teleoperator opens the drawer, moves the objects into the drawer, and closes the drawer.

These experiments used two different subsets of the Open X-Embodiment (OXE) for $\mathcal{D}_q$: Magic Soup++ (MS++) and Custom OXE (COXE). MS++ is a collection of 27 datasets that was taken directly from the OpenVLA paper [5].

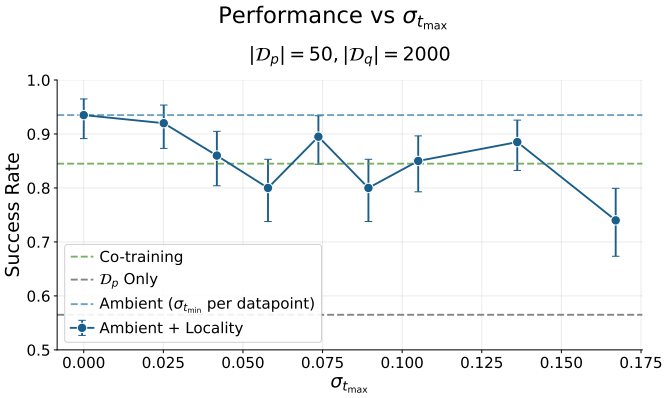
Custom OXE is a collection of 48 datasets. All 27 datasets from MS++ were included in COXE. Afterward, any dataset that met the following criteria was included:

- Robot morphology: single arm or mobile manipulator
- Action space: EEF position or EEF velocity
- Has a known control frequency
- Observation space: has proprioception and contains at least one non-depth camera
- Accessible: a few datasets that met the criteria above were not accessible, contained ambiguous features and action-space labels, or bugs. These datasets were excluded.

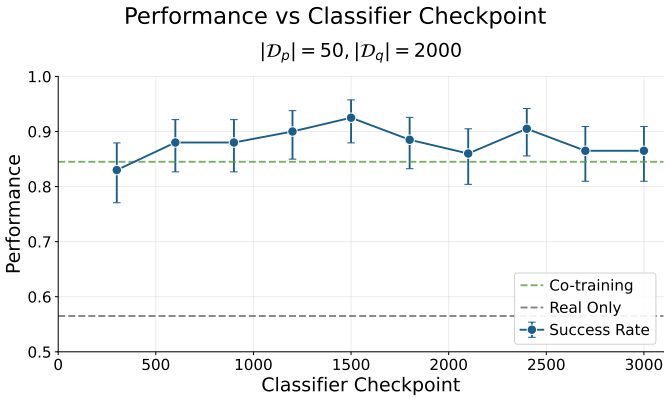
Dataset Weights: Weights for MS++ were taken from OpenVLA [5]. For COXE, we reused weights for the MS++ subset. For the remaining datasets, we assigned weights that were proportional to their size except for the two largest datasets. For those, we halved their relative weight. Following the "50/50 co-training" in previous literature [92][26], we normalize the total weight of $\mathcal{D}_q$ to be 0.5, and assign $\mathcal{D}_p$ a weight of 0.5. This is equivalent to assigning $\alpha = 0.5$ in the co-training case. In the unweighted case, the weight of each dataset is proportional to its relative size. See Figure 19 for a visualization.



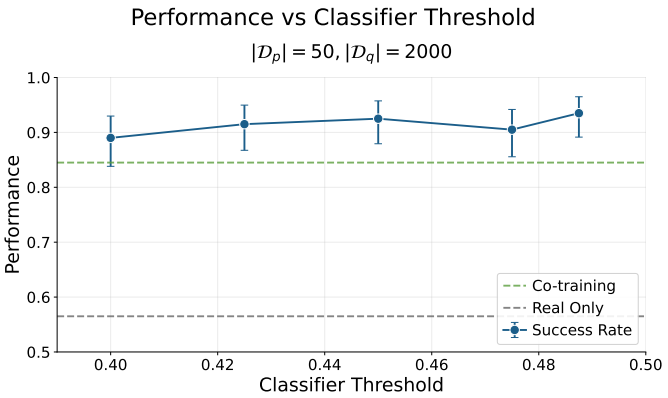
(a) Performance vs $\sigma_{t_{\min}}$ (per dataset).



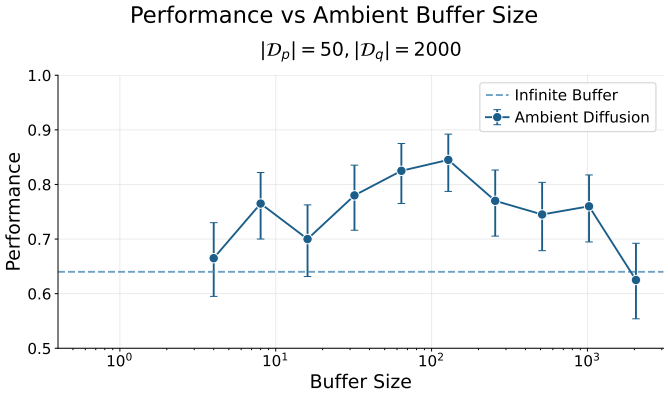
(b) Performance vs $\sigma_{t_{\max}}$.



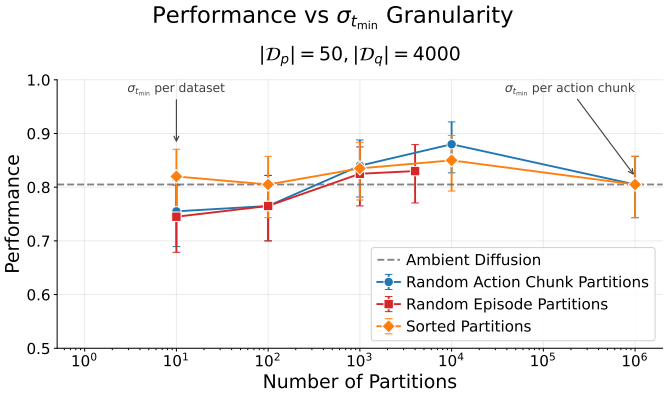
(c) Sensitivity to classifier training duration.



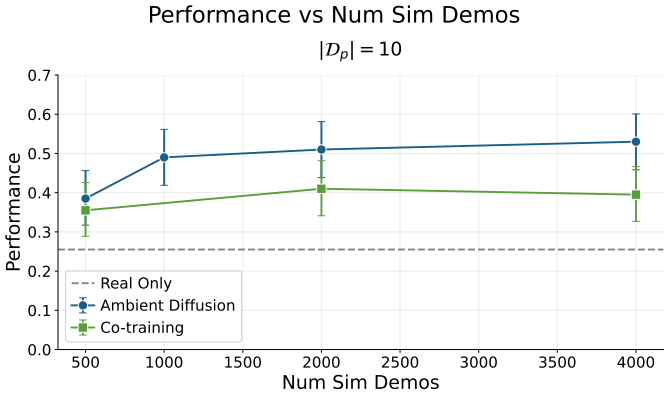
(d) Sensitivity to classifier threshold τ.



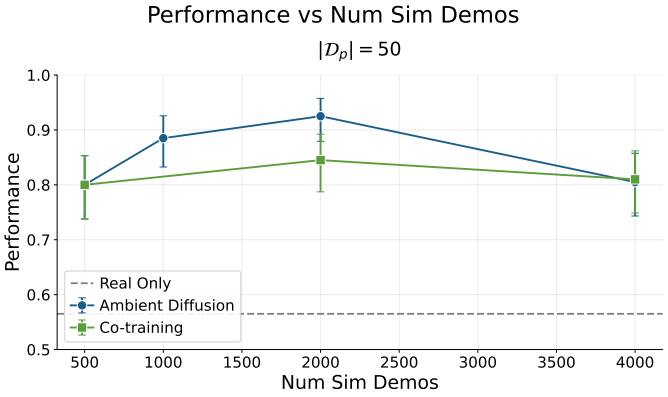
(e) Sensitivity to the Ambient loss buffer (Appendix G0a).



(f) Performance vs granularity of $t_{\min}$ labels.



(g) Performance vs $|\mathcal{D}_q|$ scale ($|\mathcal{D}_p| = 10$).



(h) Performance vs $|\mathcal{D}_q|$ scale ($|\mathcal{D}_p| = 50$).

Fig. 16: **Sim-and-real co-training ablations.** Eight ablations covering the key design decisions of Ambient Diffusion Policy. See Appendix R for the per-panel analysis.

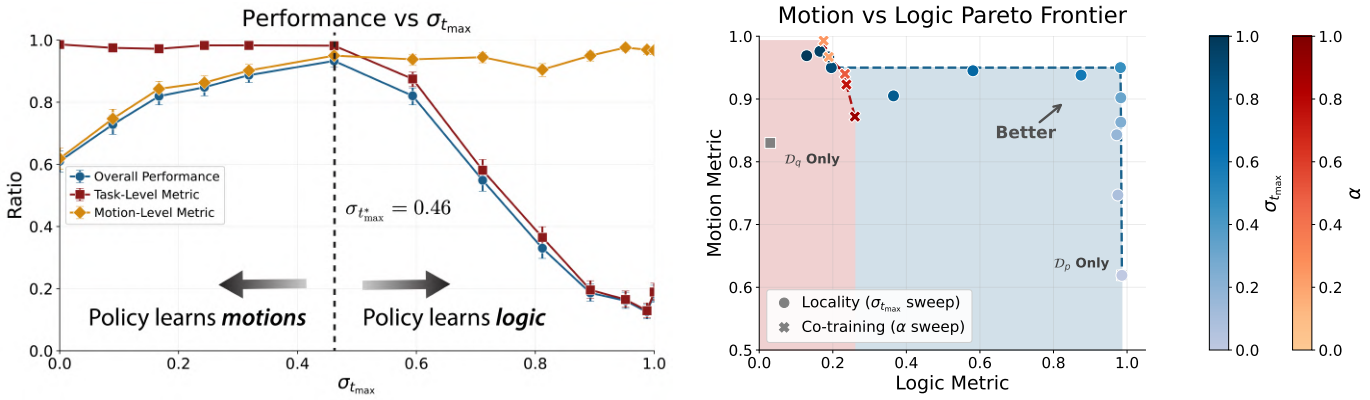
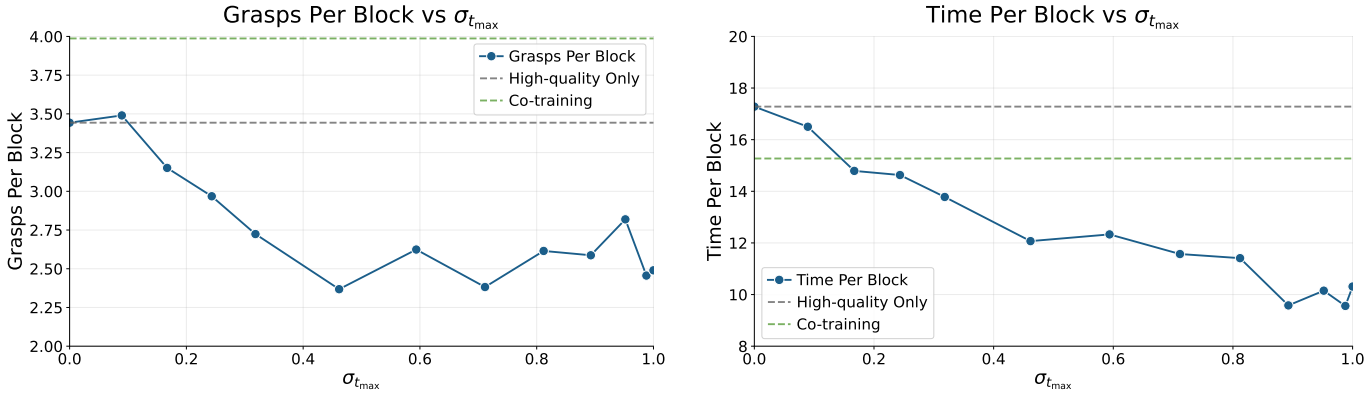


Fig. 17: **Left: Performance metrics vs** $\sigma_{t_{\max}}$. The logic metric plateaus before $\sigma_{t^*_{\max}} = 0.46$ and deteriorates rapidly afterwards; the opposite is true for the motion metric. Thus, the policy learns *local* motion-level features for $t \in [0, t^*_{\max})$ and *global* logic-level features for $t \in (t^*_{\max}, T]$. **Right:** Unlike co-training, Ambient achieves a near-optimal trade-off between logic and motion metrics.



(a) Grasp attempts per block vs $\sigma_{t_{\max}}$.

(b) Time per block vs $\sigma_{t_{\max}}$.

Fig. 18: **Additional motion-level metrics for the block sorting experiment (Section N). (a)** Grasp attempts per block improves as $\sigma_{t_{\max}}$ increases but plateaus after $\sigma_{t^*_{\max}} = 0.46$. **(b)** Time per block improves as $\sigma_{t_{\max}}$ increases.

Annotating $t_{\min}$: For each sub-dataset in COXE (e.g. DROID [92]), we train a classifier to discern samples from $\mathcal{D}_p$ and the sub-dataset. We use this classifier to assign a single $t_{\min}$ value to all action chunks within this sub-dataset. This required us to train 48 classifiers for the 48 sub-datasets in COXE. Although in principle, we could have accomplished this using a single classifier $c_\phi(A_t, t, l)$, where $l \in \{0,1\}^{48}$ is a one-hot encoding indicating the source dataset of A_t. Figures 19 and 20 show the distribution of $t_{\min}$ assigned to the sub-datasets of COXE for both the table cleaning and the tower building task.

Annotating $t_{\max}$: We simply set $t_{\max} = 10$ for the table cleaning and $t_{\max} = 5$ for tower building. We did not perform a sweep. These values are unlikely to be optimal, especially since adding locality showed mixed results for table cleaning. Lowering $t_{\max}$ in the tower building experiments appears to help.

Daras et al. [19] present a method to automatically label $t_{\max}$ using a similar classifier-based approach. Their approach assigned $t_{\max} = 0$ to nearly all datasets in MS++ and COXE. A full hyperparameter sweep on real hardware would have been prohibitively time-consuming. Hence, we selected values that are small enough to be conservative, yet large enough for a meaningful evaluation of locality.

Table Cleaning Evaluation: During each trial, 3 random unseen objects were placed on the table. The robot's task completion is measured according to the rubric in Table XI. The time limit is 90s. Each policy was evaluated on 20 trials.

The results include a minor camera resolution bug that *lowers* the policy performance across all table cleaning experiments. The image data was collected at 640 by 320 pixels and resized to 224 by 224 for training. Due to a bug at inference time, the image streams were 128 by 128 and upscaled to 224 by 224 for the policy; thus, the images during inference were effectively lower resolution than the images during training.

Tower Building Evaluation: The performance of each trial is the number of correct blocks placed before the robot topples the tower or requires human intervention. Correct indicates that the blocks follow the correct alternating direction and

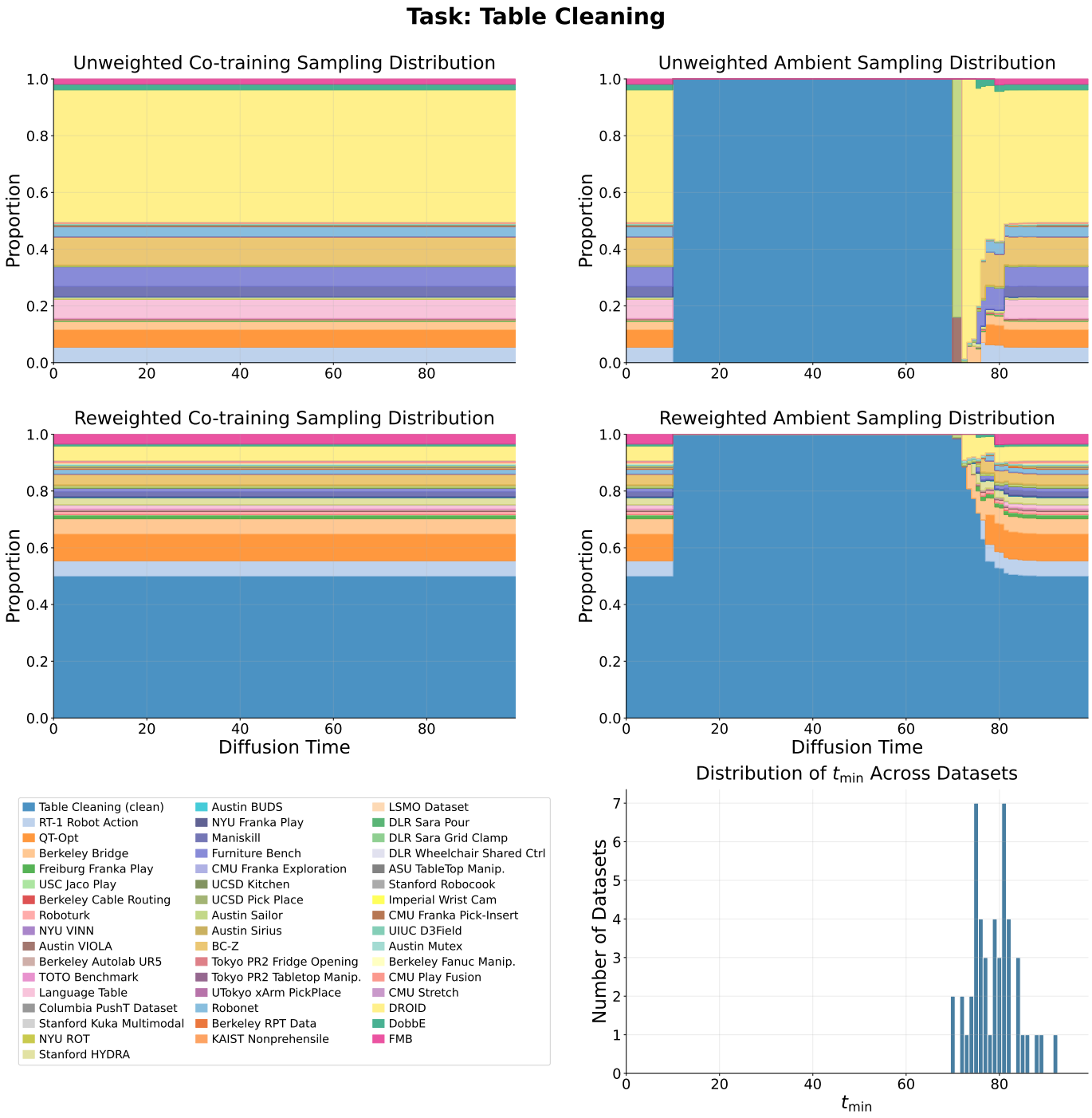


Fig. 19: **Table Cleaning: Dataset re-weighting and t_{min} visualization.** These plots show the sampling ratio for $\mathcal{D}_p$ (shown in blue) and each dataset in COXE at different diffusion times for the table cleaning task. **Left:** Co-training uses the same sampling ratio at all diffusion times. **Right:** The sampling ratio for each dataset varies at each diffusion time. Notably, at intermediate diffusion times, only data from $\mathcal{D}_p$ can be used (the "Donut paradox" from Daras et al. [19]). The histogram shows the distribution of t_{min} values assigned by the classifiers to each dataset in COXE.

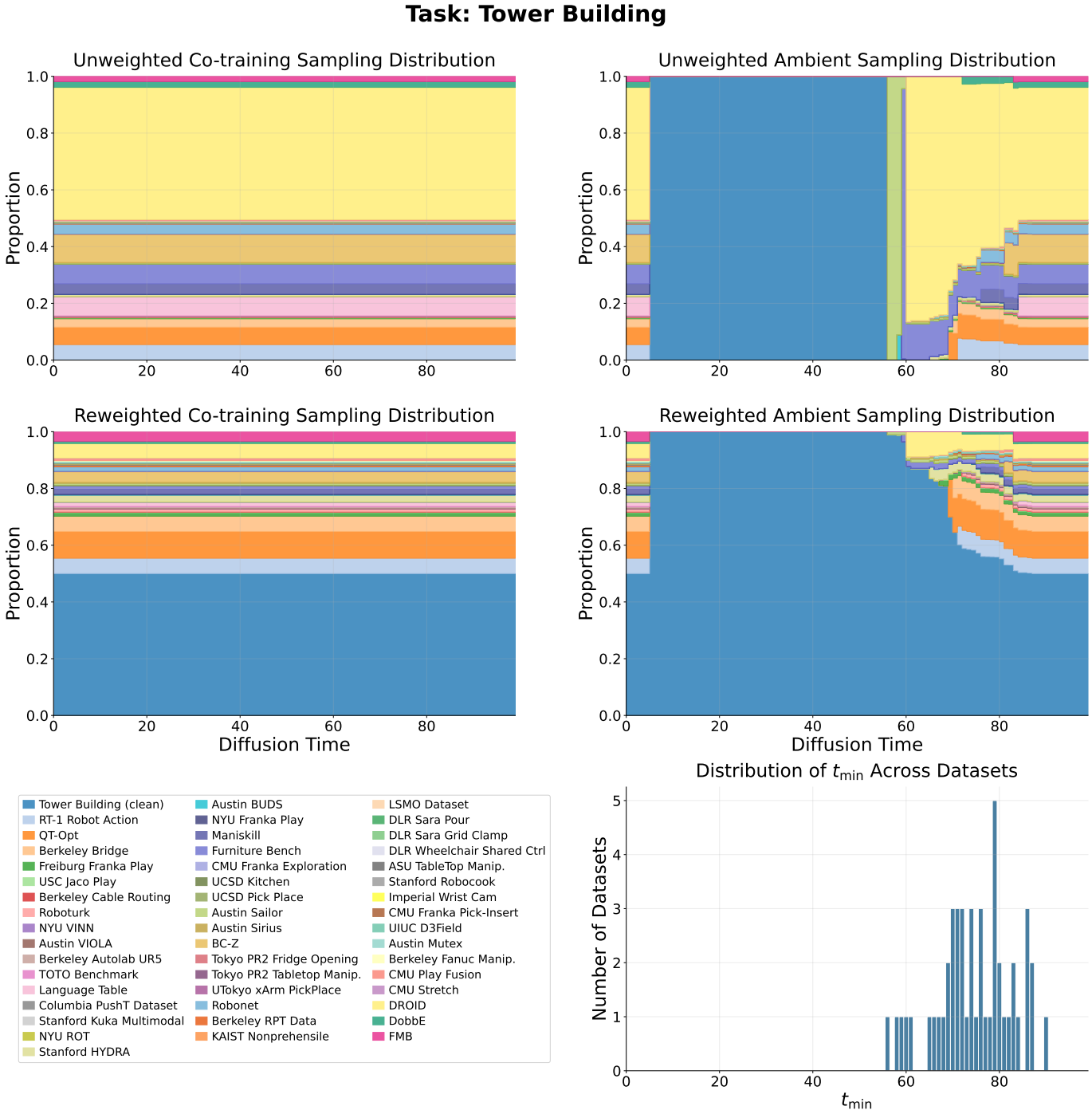


Fig. 20: **Tower Building: Dataset re-weighting and t_{min} visualization.** See Figure 19 for the equivalent plot descriptions.

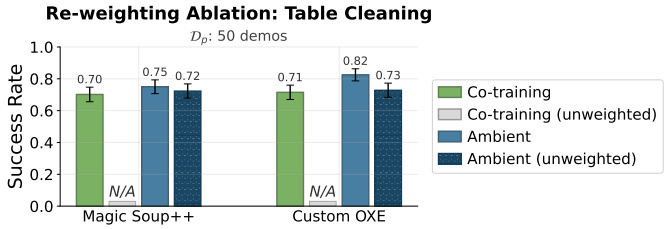


Fig. 21: **Ambient is significantly less sensitive to dataset re-weighting than co-training.** The Ambient Diffusion Policy performed at most 9% worse without re-weighting. The co-trained policies were too dangerous to evaluate without re-weighting.

color shown in Figure 1f. Each policy was evaluated on 20 trials.

The camera resolution bug from the table cleaning experiments was resolved for the tower building experiments.

U. OXE Ablations

1) Qualitative Performance: The qualitative gap between the Ambient and co-trained policies is larger than the task completion metric would suggest. The 90s timeout artificially inflates the task completion rates of weak policies: they can fail repeatedly without penalty as long as they eventually succeed. To capture this, we recorded the number of grasp attempts and time required per object for each policy. Tables XII and XIII show that the Ambient Diffusion Policies require 25-40% fewer grasps and time per object than their co-trained counterparts.

Figure 21 shows the dataset re-weighting ablation, which is discussed in the main body (Section V-B).

V. Implementation

To finetune the base policies, we reduce the learning rate by 10x to 20x. Checkpoints are saved frequently since finetuning a model for too long can degrade performance. In the simulated experiments, we evaluate every checkpoint and report the performance of the best policy. In the real-world experiments, we perform a preliminary evaluation of each checkpoint with 5-10 trials, and perform a full set of trials on the best checkpoint.

Figure 22 shows the controlled finetuning comparison, which is discussed in the main body (Section O).

W. Table Cleaning (OXE)

Figure 23 compares finetuning the best co-trained and Ambient OXE policies on $\mathcal{D}_p$ in the table cleaning task; neither produced a statistically significant gain. The results are consistent with the controlled experiments in Section O.

Ambient Diffusion Policy does not yet handle distribution shifts in the observation space. We explore two initial attempts in Appendix W. The classifier-based annotation method for $t_{\min}$ and $t_{\max}$ is theoretically grounded, but does not always outperform a (costly) hyperparameter sweep. Better annotation methods could improve performance. Lastly, our method is more computationally expensive than data filtering since it trains on more data, but in most cases, its benefits are worthwhile.

Ambient Diffusion Policy requires the user to partition their data into $\mathcal{D}_p$ and $\mathcal{D}_q$. The algorithm itself poses no restrictions on this partitioning, but it is an important design decision for downstream performance. This paper defines $\mathcal{D}_p$ as data from the target task and environment, but the right choice depends on the goal. For task-specific training or finetuning, our definition is appropriate; for pretraining a generalist policy, prior work suggests that $\mathcal{D}_p$ should contain diverse and high-quality demonstrations [74, 58]. We have not yet tested our method in the latter setting.

We explore two algorithms for handling suboptimality in the observation space. Other approaches exist – such as jointly denoising actions and observations. We note that "suboptimality" in the observation space can be ill-defined: an image from a different task is not necessarily suboptimal. In fact, it could increase the diversity of the dataset which is desirable.

Although Ambient Diffusion Policy only addresses action-space distribution shift, its results are already strong. Whether observation-space shift presents a fundamental problem and possible solutions remain open directions.

X. Noising the Observations

Ambient Diffusion Policy contracts distributional differences in the actions by adding noise. We explore the same solution for the observations. During training, we add noise to the observation input (images, proprioception, etc) for half of the samples using the same noise schedule as the actions. The model takes in the noise level for both the actions and the observations. At inference time, the model performs inference with clean observations. This ablation is conceptually similar to concurrent work that noises observations [93]; however, their goal was to expand the sampling distribution of a base policy for RL finetuning. Figure 24a presents the results; it did not appear to help for table cleaning.

Y. Classifier-Free Guidance

In classifier-free guidance, the model learns to denoise the actions both unconditionally (i.e. without observations) and conditionally (with observations) [94]. We use the same training implementation as [94]. At sampling-time, we use the denoiser in (81) where w is the guidance weight. We explored classifier-free guidance as a potential solution since it trains an unconditional denoiser which is unaffected by corruptions in the observations.

$$\tilde{h}_\theta(A_t, O, t) = (1+w)\cdot h_\theta(A_t, O, t) - w \cdot h_\theta(A_t, \varnothing, t) \quad (81)$$

Figure 24b shows that classifier-free guidance does not improve performance for all swept values of w.

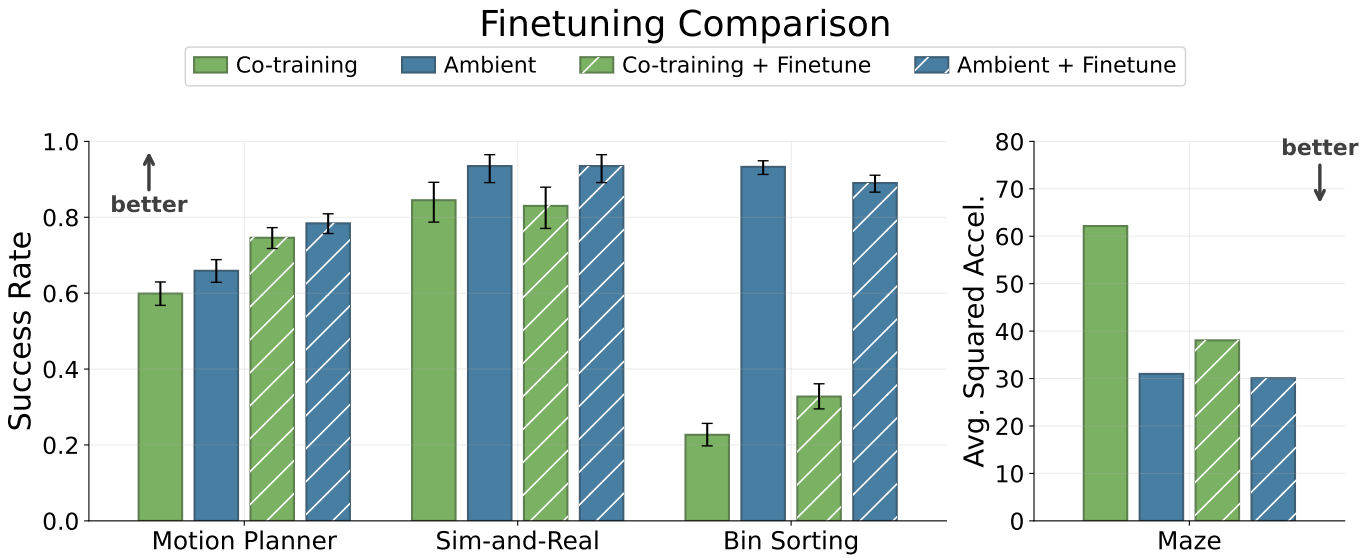


Fig. 22: **Left:** success rate of policies finetuned from different base policies. **Right:** smoothness of finetuned policies in the maze experiment. We compare smoothness instead of success rate since all policies achieved 99.0%+ success rate.

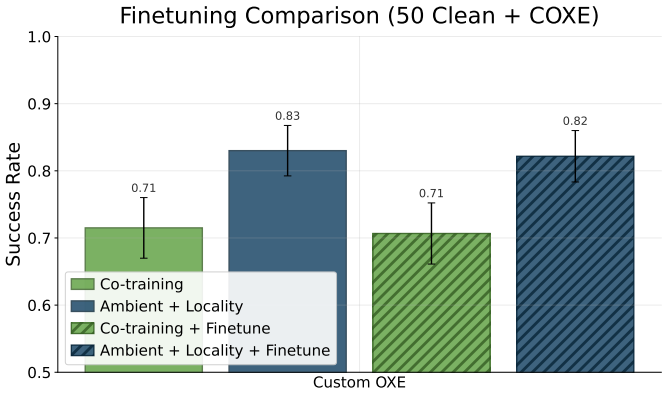
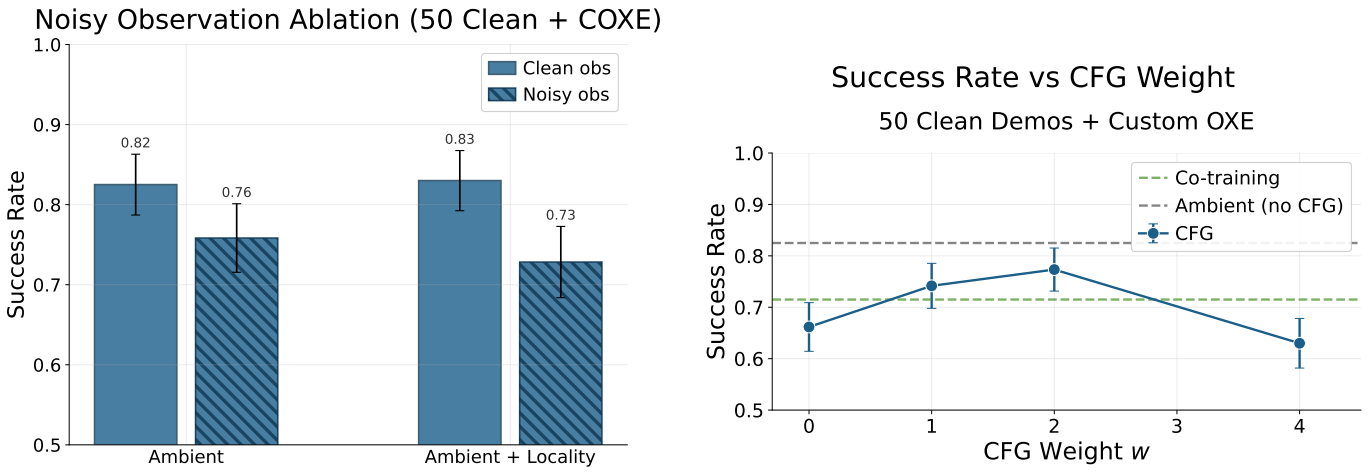


Fig. 23: Finetuning resulted in no statistically significant change in policy performance on the table cleaning task.



(a) Noising the observations during training.

(b) Classifier-free guidance vs. guidance weight w.

Fig. 24: **Algorithms for handling distribution shift in the observation space.** Neither approach produced stronger policies on the table cleaning task. **(a)** Adding Gaussian noise to the observations during training. **(b)** Classifier-free guidance [94] swept over guidance weights w.

TABLE II: Experimental configurations for the four *generality* experiments. Parameters vary for the ablations.

		Maze	**Neural Motion Planning**	**Sim-and-Real Co-training**	**Block Sorting**
Dataset Details	$\mathcal{D}_p$	GCS [65]	Trajectory Optimization [83]	Joystick Teleop [12]	VR Teleop
	$\mathcal{D}_q$	RRT [66]	BiRRT [84]	Contact-Rich Planner [72]	VR Teleop
	$\lvert\mathcal{D}_p\rvert$ **(Num. Demos.)**	50	100,000	50	50
	$\lvert\mathcal{D}_q\rvert$ **(Num. Demos.)**	5000	1,000,000	2000	200
	Nature of shift	Low-quality/ noisy trajectories	Low-quality/ noisy trajectories	Sim-to-real gap	Task mismatch
Policy I/O	**Observation space**	Proprio	Proprio	EE position Scene camera Wrist camera	EE pose (xyz, R9) Gripper state Scene camera Blocks camera Wrist camera
	Action space	x-y position	Joint angles $\mathbb{R}^7$	x-y pusher position	EE command (xyz, R9, gripper)
Parameters	$\sigma_{t^*_{\min}} \in [0,1]$	0.074 (sweep)	0.025 (sweep)	varies see Table V	N/A (sweep)
	$\sigma_{t^*_{\max}} \in [0,1]$	N/A	N/A	0 (sweep)	0.46 (sweep)
	$\alpha^* \in [0,1]$ **for co-training**	0.019 (sweep)	0.091 (sweep)	0.25 (sweep)	0.9 (sweep)
Loss	**Loss prediction target**	x_0	ϵ	x_0	ϵ
	Loss function	$\mathcal{L}_{\text{ambient}}$	$\mathcal{L}_{\text{denoising}}$	$\mathcal{L}_{\text{denoising}}$	$\mathcal{L}_{\text{denoising}}$
Eval.	**Num. trials**	1,000	1,000	200	200 (800 blocks)

TABLE III: Experimental configurations for the two *OXE scaling* experiments. Parameters vary for the ablations.

		Table Cleaning	**Tower Building**
Dataset Details	$\mathcal{D}_p$	VR Teleop	VR Teleop
	$\mathcal{D}_q$	OXE [1]	OXE [1]
	$\lvert\mathcal{D}_p\rvert$ **(Num. Demos.)**	50 & 150	35
	$\lvert\mathcal{D}_q\rvert$ **(Num. Demos.)**	1,279,774 (MS++) 1,402,606 (COXE)	1,402,606 (COXE)
	Nature of shift	Unstructured	Unstructured
Policy I/O	**Observation space**	EE pose (xyz, R9) Gripper state Table camera Drawer camera 2x Wrist camera	EE pose (xyz, R9) Gripper state Table camera Tower camera 2x Wrist camera
	Action space	Delta pose (xyz, axis-angle, gripper)	Delta pose (xyz, axis-angle, gripper)
Parameters	$\sigma_{t^*_{\min}} \in [0, 1]$	See Figure 19 (classifier)	See Figure 20 (classifier)
	$\sigma_{t^*_{\max}} \in [0, 1]$	0.167 (no sweep)	0.09 (no sweep)
	$\alpha^* \in [0, 1]$ **for co-training**	0.5 (no sweep)	0.5 (no sweep)
Loss	**Loss prediction target**	ϵ	ϵ
	Loss function	$\mathcal{L}_{\text{denoising}}$	$\mathcal{L}_{\text{denoising}}$
Eval.	**Num. trials**	20 (60 objects)	20

TABLE IV: **Maze and Neural Motion Planning results (1000 trials each).** In both settings, Ambient Diffusion Policy achieves the highest success rate and is substantially smoother than co-training.

Training Algo.	Success Rate ↑	Smoothness ↓ (Eq. (78))
Data Filtering	$57.5^{+3.1}_{-3.1}$ %	31.9
Co-train ($\alpha^* = 0.019$)	$99.4^{+0.4}_{-0.7}$ %	62.2
Ambient ($\sigma_{t^*_{\min}} = 0.074$)	$\mathbf{99.5^{+0.3}_{-0.7}}$ %	**31.0**

(a) 2D maze.

Training Algo.	Success Rate ↑	Smoothness ↓ (Eq. (78))
Data Filtering	$46.0^{+3.1}_{-3.1}$ %	**3.9**
Co-train ($\alpha^* = 0.091$)	$59.9^{+3.1}_{-3.1}$ %	42.7
Ambient ($\sigma_{t^*_{\min}} = 0.025$)	$\mathbf{65.9^{+2.9}_{-3.0}}$ %	31.4

(b) 7-DoF neural motion planning.

TABLE V: **Main planar pushing results (200 trials).** All Ambient variations outperform co-training [12].

Training Algorithm	Success Rate ↑
Data Filtering	$56.5^{+7.0}_{-7.2}$ %
Co-training [12]	$84.5^{+4.7}_{-5.8}$ %
Ambient ($t_{\min}$ per dataset)	$87.0^{+4.3}_{-5.5}$ %
Ambient ($t_{\min}$ per datapoint)	$\mathbf{93.5^{+3.0}_{-4.4}}$ %
Ambient + Locality ($\sigma_{t_{\max}} = 0.025$)	$92.0^{+3.4}_{-4.7}$ %

TABLE VI: **Main block sorting results (∼800 blocks).** Only Ambient Diffusion Policy (Locality) performs well on all metrics.

Training Algo.	Data Filtering	$\mathcal{D}_q$ Only	Co-train ($\alpha^* = 0.9$)	Locality ($\sigma_{t^*_{\max}} = 0.46$)
Logic ↑	$\mathbf{98.6^{+0.8}_{-1.5}}$ %	$3.0^{+1.6}_{-1.2}$ %	$26.0^{+3.4}_{-3.2}$ %	$98.2^{+0.8}_{-1.2}$ %
Motion ↑	$61.9^{+3.4}_{-3.5}$ %	$83.0^{+2.5}_{-2.8}$ %	$87.2^{+2.2}_{-2.5}$ %	$\mathbf{95.0^{+1.4}_{-1.7}}$ %
Success Rate ↑	$61.0^{+3.4}_{-3.5}$ %	$2.5^{+1.3}_{-1.0}$ %	$22.7^{+3.1}_{-2.9}$ %	$\mathbf{93.3^{+1.6}_{-2.0}}$ %

TABLE VII: Comparison between the 2D maze and 7-DoF neural motion planning experiments.

	2D Maze	**7-DoF NMP**
$\mathcal{D}_p$ **Source**	GCS [65]	Trajectory Optimizer [83]
$\mathcal{D}_q$ **Source**	RRT [66]	BiRRT [84]
$\lvert\mathcal{D}_p\rvert$	50	100,000
$\lvert\mathcal{D}_q\rvert$	5,000	1,000,000
Output Type	Action Chunk	Full Plan
Rollouts	Closed-Loop	Open-Loop
Penetration Threshold	0cm	2cm
Output Space	2D (x, y)	7D (Config. Space)
Start & End Goals	Uniformly Random	Around points of interest (See Appendix Q)

TABLE IX: **Sim-to-target gap along three axes**. The sim-to-target gap in the environments from [12] was designed to mimic the sim-to-real gap. See [12] for visualizations.

Gap	**Simulation**	**Target Sim**
Visual	White spectrum light	Warm spectrum light
	Different object colors	Different object colors
	Different camera pose	Different camera pose
	No shadows	Shadows
Physics	Quasistatic	Drake physics [90]
Action	Motion planning	Human teleop

TABLE X: **Block sorting ablations (200 trials, ∼800 blocks). (a)** Ablation 1 (Task-Conditioned): Ambient outperforms co-training. **(b)** Ablation 2: the Ambient Diffusion Policy performs well despite training on no logically correct samples for $t \in [0, t^*_{\max})$.

	(a) Ablation 1: Task-Conditioned		**(b) Ablation 2**
Training Algo. (# Blocks: ∼800)	Co-train ($\alpha^* = 0.5$)	Locality ($\sigma_{t^*_{\max}} = 0.89$)	Locality $\mathcal{D}_q$ only below $\sigma_{t^*_{\max}} = 0.24$
Logic ↑	$\mathbf{98.6}^{+\mathbf{0.7}}_{-\mathbf{1.1}}$ %	$98.5^{+0.7}_{-1.1}$ %	$97.9^{+0.9}_{-1.3}$ %
Motion ↑	$91.5^{+1.8}_{-2.2}$ %	$\mathbf{94.2}^{+\mathbf{1.5}}_{-\mathbf{1.8}}$ %	$93.8^{+1.6}_{-1.9}$ %
Success Rate ↑	$90.3^{+2.0}_{-2.3}$ %	$\mathbf{92.8}^{+\mathbf{1.7}}_{-\mathbf{2.0}}$ %	$91.9^{+1.8}_{-2.1}$ %

TABLE XI: **Table cleaning evaluation rubric.** The maximum score per trial is 1.0.

Subtask	**Points**
Open drawer	0.1
Clean object (per object, 3 total)	$0.8/3 \approx 0.267$
Close drawer (after table cleaned)	0.1
Max score per trial	**1.0**

TABLE XII: Average grasp attempts per object. The error bars on the **Average** row report the standard error of the mean across per-object values in each column ($n = 10$ objects).

	MS++		**COXE**	
Object	Co-train	Ambient + Locality	Co-train	Ambient + Locality
Apple	2	1	2	2
Figurine	4	2	2	2
Eraser	4	2	2	1
Play-Doh	2	2	2	2
Robot Gripper	3	2	2	2
Yarn	2	2	2	2
Camera Mount	1	1	1	2
3D Printed Object	2	2	3	1
Mustard	3	1	2	2
Garbage Bag Box	1	1	3	1
Average	2.4 ± 0.3	1.6 ± 0.2	2.1 ± 0.2	1.7 ± 0.2

TABLE XIII: Average cleaning time per object. The error bars on the **Average** row report the standard error of the mean across per-object values in each column ($n = 10$ objects).

	MS++		**COXE**	
Object	Co-train	Ambient + Locality	Co-train	Ambient + Locality
Apple	21s	9s	30s	19s
Figurine	44s	20s	30s	20s
Eraser	39s	16s	18s	8s
Play-Doh	20s	20s	19s	20s
Robot Gripper	31s	19s	19s	20s
Yarn	22s	19s	20s	20s
Camera Mount	12s	8s	8s	20s
3D Printed Object	19s	20s	29s	8s
Mustard	33s	9s	18s	19s
Garbage Bag Box	8s	8s	30s	7s
Average	24.9 ± 3.6s	14.8 ± 1.7s	22.1 ± 2.3s	16.1 ± 1.8s