Octo: An Open-Source Generalist Robot Policy

Octo Model Team

Dibya Ghosh*,[1] Homer Walke*,[1] Karl Pertsch*,[1,2] Kevin Black*,[1] Oier Mees*,[1]

Sudeep Dasari[3] Joey Hejna[2] Tobias Kreiman[1] Charles Xu[1] Jianlan Luo[1] You Liang Tan[1]
Dorsa Sadigh[2] Chelsea Finn[2] Sergey Levine[1]

[1]UC Berkeley [2]Stanford [3]Carnegie Mellon University

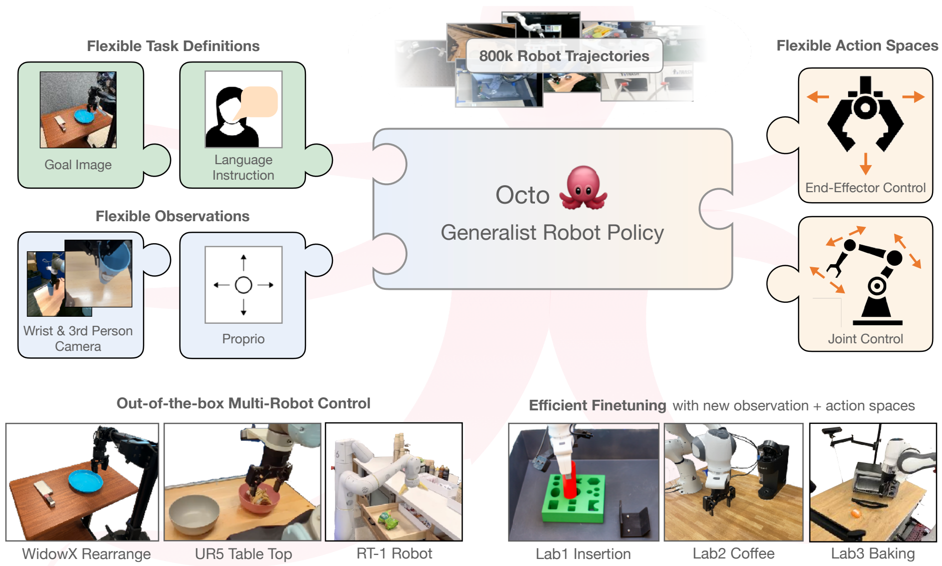

Fig. 1: We introduce Octo, an open-source, generalist policy for robotic manipulation. Octo is a transformer-based policy pretrained on 800k diverse robot episodes from the Open X-Embodiment dataset [1]. It supports flexible task and observation definitions and can be quickly finetuned to new observation and action spaces.

Abstract— **Large policies pretrained on diverse robot datasets have the potential to transform robotic learning: instead of training new policies from scratch, such generalist robot policies may be finetuned with only a little in-domain data, yet generalize broadly. However, to be widely applicable across a range of robotic learning scenarios, environments, and tasks, such policies would need to handle diverse sensors and action spaces, accommodate a variety of commonly used robotic platforms, and finetune readily and efficiently to new domains.**

Large policies pretrained on diverse robot datasets have the potential to transform robotic learning: instead of training new policies from scratch, such generalist robot policies may be finetuned with only a little in-domain data, yet generalize broadly. However, to be widely applicable across a range of robotic learning scenarios, environments, and tasks, such policies would need to handle diverse sensors and action spaces, accommodate a variety of commonly used robotic platforms, and finetune readily and efficiently to new domains. In this work, we aim to lay the groundwork for developing open-source, widely applicable, generalist policies for robotic manipulation. As a first step, we introduce Octo, a large transformer-based policy trained on 800k trajectories from the Open X-Embodiment dataset, the largest robot manipulation dataset to date. It can be instructed via language commands or goal images and can be effectively finetuned to robot setups with new sensory inputs and action spaces within a few hours on standard consumer GPUs. In experiments across 7 robotic platforms, we demonstrate that Octo serves as a versatile policy initialization that can be effectively finetuned to new observation and action spaces. We also perform detailed ablations of design decisions for the Octo model, from architecture to training data, to guide future research on building generalist robot models.

*Lead authors, ordered alphabetically

I. INTRODUCTION

The common approach for robotic learning is to train policies on datasets collected for the specific robot and task at hand. Learning from scratch in this way requires significant data collection effort for each task, and the resulting policies usually exhibit only narrow generalization. In principle, collected experience from other robots and tasks offers a possible solution, exposing models to a diverse set of robotic control problems that may improve generalization and performance on downstream tasks. However, even as general-purpose models become ubiquitous in natural language [2], [3]) and computer vision [4], [5], it has proven challenging to build the analogous "general-purpose robot model" that can control many robots for many tasks. Training a unified con-

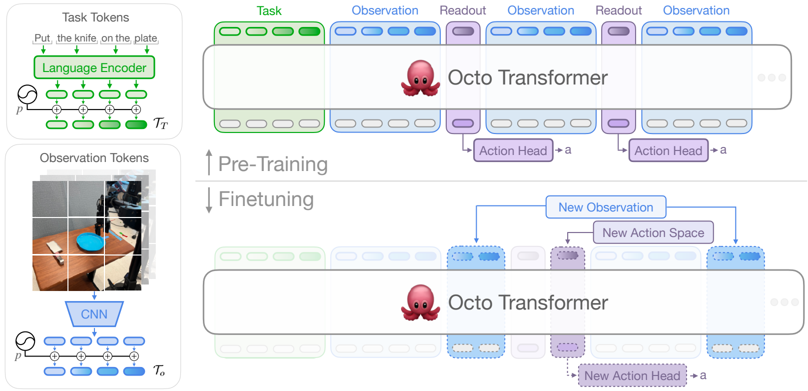

Fig. 3: **Model architecture. Left**: Octo tokenizes task descriptions (green) and input observations (blue) using a pretrained language model and a lightweight CNN, respectively. **Top**: The transformer backbone processes the sequence of task and observation tokens and produces readout tokens (purple) that get passed to output heads to produce actions. **Bottom**: The block-wise attention structure of the transformer allows us to add or remove inputs and outputs during finetuning: for example, we can add new observations (blue, dashed) or action spaces (purple, dashed) without modifying any pretrained parameters.

trol policy in robotics presents unique challenges, requiring handling different robot embodiments, sensor setups, action spaces, task specifications and environments.

Towards this direction, several works have proposed robotic foundation models that directly map robot observations to actions and provide zero-shot or few-shot generalization to new domains and robots. We broadly refer to these models as "generalist robot policies" (GRPs), emphasizing their ability to perform low-level visuomotor control across tasks, environments, and robotic systems [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. For example, the GNM model [17] generalizes across different robotic navigation scenarios, the RoboCat model [7] handles different robot embodiments for goal-conditioned tasks, and the RT-X model [1] performs language-conditioned manipulation across five robot embodiments.

Although these models represent significant steps toward a true "general-purpose robot model," they have been limited in multiple important aspects: they typically constrain downstream users to a pre-defined and often restrictive set of input observations, e.g., a single camera stream; they lack support for effective finetuning to new domains; and importantly, the largest of these models are not available to the general public.

Our primary contribution is Octo, a transformer-based policy pretrained on the largest robot manipulation dataset to date: 800k robot trajectories from the Open X-Embodiment dataset [1]. Octo is the first GRP that can be effectively finetuned to new observations and action spaces and the first generalist robot manipulation policy that is fully open-source, including the training pipeline, model checkpoints, and data. Finally, while the individual components that comprise Octo — a transformer backbone, support for both language and goal image specification, and a diffusion head to model expressive action distributions — have been discussed in prior work, the particular combination of these components into a powerful generalist robot policy is unique and novel.

We demonstrate through extensive experiments on 7 robots across 4 institutions that our combined system leads to state-of-the-art performance for zero-shot multi-robot control and that Octo can be used as an effective initialization for finetuning to unseen robot setups with new observation and action spaces.

II. The Octo Model

In this section, we describe the Octo model, our open-source generalist robot policy that can be adapted to new robots and tasks — including new sensory inputs and action spaces — via finetuning. We discuss the key design decisions, training objectives, training dataset, and infrastructure. The design of the Octo model emphasizes flexibility and scale: it supports a variety of commonly used robots, sensor configurations, and actions while providing a generic and scalable recipe that can be trained on large amounts of data. It also supports natural language instructions, goal images, observation histories, and multi-modal, chunked action prediction via diffusion decoding [18]. Furthermore, we designed Octo specifically to enable efficient finetuning to new robot setups, including robots with different action spaces and different combinations of cameras and proprioceptive information. This design was selected to make Octo a flexible and broadly applicable generalist robot policy that can be utilized for a variety of downstream robotics applications and research projects.

A. Architecture

At its core, Octo is a transformer-based policy π. It consists of three key parts: **input tokenizers** that transform language instructions ℓ, goals g, and observation sequences $o_1, \ldots, o_H$ into tokens $\left[\mathcal{T}_l, \mathcal{T}_g, \mathcal{T}_o\right]$ (Fig. 3, left); a **transformer backbone** that processes the tokens and produces embeddings $e_l, e_g, e_o = T(\mathcal{T}_l, \mathcal{T}_g, \mathcal{T}_o)$ (Fig. 3, top); and **readout heads** $R(e)$ that produce the desired outputs, i.e., actions a.

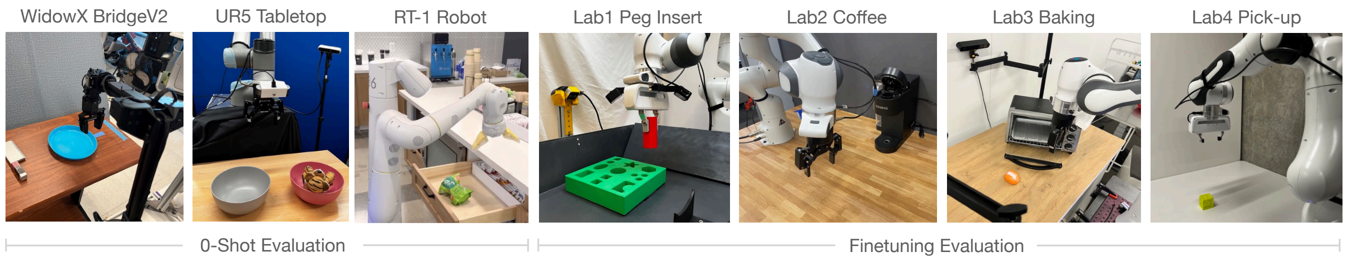

Fig. 4: **Evaluation Tasks.** We evaluate Octo on 7 real robot setups across 4 institutions. Our evaluations capture diverse object interactions (e.g., "WidowX BridgeV2"), long task horizons (e.g., "Lab2 Coffee") and precise manipulation (e.g., "Lab1 Insertion"). We evaluate Octo's capabilities to control robots in environments from the pretraining data out-of-the-box and to efficiently finetune to new tasks and environments with small target domain datasets. We also test finetuning with new observations (force-torque inputs for "Franka Insertion") and action spaces (joint position control in "Lab4 Pick-Up").

Task and observation tokenizers: We convert task definitions (e.g., language instructions ℓ and goal images g) and observations o (e.g., wrist and third-person camera streams) into a common "tokenized" format using modality-specific tokenizers (see Fig. 3, left):

- **Language inputs** are tokenized, then passed through a pretrained transformer that produces a sequence of language embedding tokens. We use the `t5-base` (111M) model [19].
- **Image observations and goals** are passed through a shallow convolution stack, then split into a sequence of flattened patches [20].

We assemble the input sequence of the transformer by adding learnable position embeddings p to task and observation tokens and then arranging them sequentially $[\mathcal{T}_T, \mathcal{T}_{o,1}, \mathcal{T}_{0,2}, \dots]$.

Transformer backbone and readout heads: Once the inputs have been cast to a unified token sequence, they are processed by a transformer (see Fig. 3, top). This is similar to prior works that train transformer-based policies on sequences of observations and actions [21], [22]. The attention pattern of the Octo transformer is blockwise masked: observation tokens can only attend causally to tokens from the same or earlier time steps $\mathcal{T}_{o,0:t}$ as well as task tokens $\mathcal{T}_T$ (green). Tokens corresponding to non-existing observations are fully masked out (e.g., a dataset without language instructions). This modular design enables us to add and remove observations or tasks during finetuning (see below). In addition to these input token blocks, we insert learned *readout tokens* $\mathcal{T}_{R,t}$ (purple). A readout token at $\mathcal{T}_{R,t}$ attends to observation and task tokens before it in the sequence, but is not attended to by *any* observation or task token — hence, they can only passively read and process internal embeddings without influencing them. An action head that implements the diffusion process is applied to the embeddings for the readout tokens. This head predicts a chunk of consecutive actions, similar to prior work [18].

Our design allows us to flexibly add new task and observation inputs or action output heads to the model during downstream finetuning. When adding new tasks, observations, or loss functions downstream, we can wholly retain the pretrained weights for the transformer, only adding new positional embeddings, a new lightweight encoder, or the parameters of the new head as necessitated by the change in specification (see Fig. 3, bottom).

This flexibility is crucial to make Octo a truly "generalist" model: since we cannot cover all possible robot sensor and action configurations during pretraining, being able to adapt Octo's inputs and outputs during finetuning makes it a versatile tool for the robotics community. Prior model designs that use standard transformer backbones or fuse visual encoders with MLP output heads lock in the type and order of inputs expected by the model. In contrast, switching the observation or task for Octo *does not* require re-initializing most of the model.

B. Training data

We train Octo on a mixture of 25 datasets from the Open X-Embodiment Dataset [1], a diverse collection of robot learning datasets. Our training mixture includes data from a variety of robot embodiments, scenes, and tasks. These datasets are heterogeneous not just in terms of the robot type, but also in the sensors (e.g., including or not including wrist cameras) and labels (e.g., including or not including language instructions). See Appendix C for the detailed mixture.

C. Training objective

We use a conditional diffusion decoding head to predict continuous, multi-modal action distributions [23], [18]. Importantly, only one forward pass of the transformer backbone is performed per action prediction, after which the multi-step denoising process is carried out entirely within the small diffusion head. We found this policy parameterization to outperform policies trained with MSE action heads or discretized action distributions [10] in both zero-shot and finetuning evaluations. To generate an action, we sample a Gaussian noise vector $x^K \sim \mathcal{N}(0, I)$ and apply K steps of denoising with a learned denoising network $\epsilon_\theta(x^k, e, k)$ that is conditioned on the output x^k of the previous denoising step, the step index k, and the output embedding e of the transformer action readout:

$$x^{k-1} = \alpha(x^k - \gamma\epsilon_\theta(x^k, e, k) + \mathcal{N}(0, \sigma^2 I)). \tag{1}$$

	Lab1 Insertion*	Lab2 Coffee	Lab3 Baking	Lab4 Pick-Up†	Average
ResNet+Transformer Scratch	10%	45%	25%	0%	20%
VC-1 [24]	5%	0%	30%	0%	9%
Octo (Ours)	**70%**	**75%**	**50%**	**60%**	**64%**

TABLE I: **Finetuning Evaluation.** Octo enables data-efficient finetuning to new domains and out-performs training from scratch as well as state-of-the-art pretrained visual representations. Each domain uses $\sim$ 100 target demonstrations and the same finetuning hyperparameters. In each domain, success rates are averaged over 20 trials. $*$: New observation input (force-torque proprioception). †: New action space (joint position control).

III. Experiments

Our experiments provide an empirical analysis of Octo, evaluating its ability to serve as a general robotic foundation model across several axes:

1) Can Octo control multiple robot embodiments and solve language and goal tasks out of the box?
2) Do Octo weights serve as a good initialization for data-efficient finetuning to new tasks and robots, and does it improve over training from scratch and commonly used pretrained representations?
3) Which design decisions in Octo matter most for building generalist robot policies?

Evaluation setups: We evaluate Octo's capabilities across a representative spectrum of 7 robot learning setups at 4 institutions (see Fig. 4). We test Octo's ability to control different robots out-of-the-box ("zero-shot") for language and goal image tasks using robot setups that match the pretraining data, where all robots are controlled with delta end-effector control actions and the observation spaces are RGB images. We also evaluate Octo for data-efficient finetuning to new environments and tasks, including with new observations (force-torque inputs in "Lab1 Insertion") and new action spaces (joint control in "Lab4 Pick-Up").

Comparisons: We compare Octo's ability to control multiple robots out-of-the-box to the best openly available generalist robot policy, **RT-1-X** [1], using the released checkpoint. Similar to Octo, RT-1-X is pretrained on the Open X-Embodiment robot dataset and aims to control multiple robots zero-shot, thus providing a natural point of comparison. We also compare the zero-shot capabilities of Octo to **RT-2-X**, a 55 billion parameter vision-language model finetuned on the Open X-Embodiment dataset to produce robot actions. We further compare Octo's performance as a policy initialization for data efficient finetuning to two common approaches: (1) training on the target domain demonstrations *from scratch* and (2) using pretrained visual representations.

A. Octo Controls Multiple Robots Out-of-the-Box

We compare the zero-shot manipulation capability of Octo, RT-1-X, and RT-2-X in Fig. 5. We evaluated on a variety of task types including picking and placing, wiping a table with a cloth, and opening and closing drawers. For each robot, we selected two language tasks and performed 10 trials per task with varying initial conditions. While all methods were able to solve a diverse range of tasks in the pretraining environments, we found that on average Octo had a 29% higher success rate than RT-1-X (35M parameters). For the WidowX and RT-1 Robot evaluations, we also compared to RT-2-X (55 billion parameters) [9] and found that Octo performed similarly.

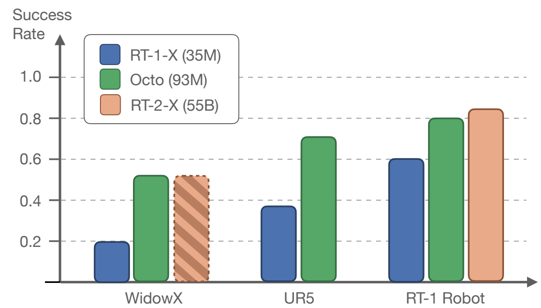

Fig. 5: **Zero-Shot Evaluation.** Out-of-the-box, Octo can control multiple robots in environments from the pretraining data, outperforming RT-1-X [1], the current best openly available generalist robot policy across three different robot embodiments and setups.

B. Octo Enables Data-Efficient Learning in New Domains

We report data-efficient finetuning results to new domains in Table I. We find that finetuning Octo leads to better policies than starting from scratch or with the pretrained VC-1 weights. On average across the four evaluation setups (detailed in Appendix F), Octo outperforms the next best baseline by 43%.

IV. Discussion and Future Work

We introduced Octo, a large transformer-based policy pretrained on the largest robot manipulation dataset to date, 800k robot trajectories. We demonstrated that Octo can solve a variety of tasks out-of-the-box and showed how Octo's compositional design enables finetuning to new inputs and action spaces, making Octo a versatile initialization for a wide range of robotic control problems. Apart from the model itself, we have released our full training and finetuning code, alongside tools that make it easier to train on large robot datasets. While Octo represents a step towards building generalist robot policies that work out-of-the-box on diverse robot setups, there remains work to improve the model, including better language conditioning, improved support for wrist cameras, and incorporating data beyond optimal demonstrations. We hope that Octo offers a simple launchpad for researchers and practitioners to access larger robotic datasets and leverage pretrained robotics models for efficient learning of new tasks and broad generalization.

REFERENCES

[1] Open X-Embodiment Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui, "Open X-Embodiment: Robotic learning datasets and RT-X models," https://arxiv.org/abs/2310.08864, 2023.

[2] OpenAI, "GPT-4 Technical Report," Mar. 2023.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022.

[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment Anything," Apr. 2023.

[6] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *Transactions on Machine Learning Research*, 2022.

[7] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju *et al.*, "Robocat: A self-improving foundation agent for robotic manipulation," *arXiv preprint arXiv:2306.11706*, 2023.

[8] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[9] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *7th Annual Conference on Robot Learning*, 2023.

[10] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[11] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://arxiv.org/abs/2306.14846

[12] S. AI. (2023) Introducing scale's automotive foundation model. [Online]. Available: https://scale.com/blog/afm1

[13] Wayve. (2023) Lingo: Natural language for autonomous driving. [Online]. Available: https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/

[14] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, "Gaia-1: A generative world model for autonomous driving," 2023.

[15] J. H. Yang, D. Sadigh, and C. Finn, "Polybot: Training one policy across robots while embracing variability," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=HEIRj51lcS

[16] V. Kumar, R. Shah, G. Zhou, V. Moens, V. Caggiano, A. Gupta, and A. Rajeswaran, "Robohive: A unified framework for robot learning," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. [Online]. Available: https://openreview.net/forum?id=0H5fRQcpQ7

[17] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233.

[18] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[21] P. Wu, A. Majumdar, K. Stone, Y. Lin, I. Mordatch, P. Abbeel, and A. Rajeswaran, "Masked trajectory models for prediction, representation, and control," *International Conference on Machine Learning*, 2023.

[22] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik, "Robot learning with sensorimotor pre-training," *Conference on Robot Learning*, 2023.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[24] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *arXiv preprint arXiv:2303.18240*, 2023.

[25] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[26] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[27] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[28] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 885–897.

[29] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[30] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *Advances in neural information processing systems*, vol. 31, 2018.

[31] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: Robot manipulation with multimodal prompts," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 14 975–15 022. [Online]. Available: https://proceedings.mlr.press/v202/jiang23b.html

[32] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[33] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[34] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-

task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[35] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn *et al.*, "Open-world object manipulation using pre-trained vision-language models," *arXiv preprint arXiv:2303.00905*, 2023.

[36] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.

[37] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "Genaug: Retargeting behaviors to unseen situations via generative augmentation," *arXiv preprint arXiv:2302.06671*, 2023.

[38] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[39] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[40] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[41] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[42] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.

[43] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Chiang Mai, Thailand, 2023.

[44] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," *arXiv preprint arXiv:2310.10639*, 2023.

[45] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," *arXiv preprint arXiv:2207.09450*, 2022.

[46] T. Kwon, N. Di Palo, and E. Johns, "Language models as zero-shot trajectory generators," *arXiv preprint arXiv:2310.11604*, 2023.

[47] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[48] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," *arXiv preprint arXiv:2310.07896*, 2023.

[49] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

[50] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[51] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[52] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[53] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[54] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.

[55] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer, "Tartandrive: A large-scale dataset for learning off-road dynamics models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2546–2552.

[56] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, "Scaling up multi-task robotic reinforcement learning," in *5th Annual Conference on Robot Learning*, 2021.

[57] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang, "Scaling data-driven robotics with reward sketching and batch reinforcement learning," *arXiv preprint arXiv:1909.12200*, 2019.

[58] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*. PMLR, 2018, pp. 879–893.

[59] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[60] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[61] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine, "Bridgedata v2: A dataset for robot learning at scale," 2023.

[62] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[63] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, vol. 3, p. 5, 2023.

[64] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," *arXiv preprint arXiv:2309.01918*, 2023.

[65] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, "Latent plans for task agnostic offline reinforcement learning," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[66] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[67] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," 2023.

[68] S. Belkhale, Y. Cui, and D. Sadigh, "Hydra: Hybrid robot actions for imitation learning," *arxiv*, 2023.

[69] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, 2023.

[70] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," in *Robotics: Science and Systems*, 2023.

[71] R. Shah, R. Martín-Martín, and Y. Zhu, "MUTEX: Learning unified policies from multimodal task specifications," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=PwqiqaaEzJ

[72] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu, "Learning and retrieval from prior data for skill-based imitation learning," in *Conference on Robot Learning (CoRL)*, 2022.

[73] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "RoboTurk: A crowdsourcing platform for robotic skill learning through imitation," *CoRR*, vol. abs/1811.02790, 2018. [Online]. Available: http://arxiv.org/abs/1811.02790

[74] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, C. Finn, and A. Gupta, "Train offline, test online: A real robot learning benchmark," 2023.

[75] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu, "Robot learning

on the job: Human-in-the-loop autonomy and learning during deployment," in *Robotics: Science and Systems (RSS)*, 2023.

[76] L. Y. Chen, S. Adebola, and K. Goldberg, "Berkeley UR5 demonstration dataset," https://sites.google.com/view/berkeley-ur5/home.

[77] S. Saxena, M. Sharma, and O. Kroemer, "Multi-resolution sensing for real-time control with vision-language models," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=WuBv9-IGDUA

[78] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," 2023.

[79] X. Zhu, R. Tian, C. Xu, M. Ding, W. Zhan, and M. Tomizuka, "Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot," 2023.

[80] Z. J. Cui, Y. Wang, N. M. M. Shafiullah, and L. Pinto, "From play to policy: Conditional behavior generation from uncurated robot data," *arXiv preprint arXiv:2210.10047*, 2022.

[81] G. Yan, K. Wu, and X. Wang, "ucsd kitchens Dataset," August 2023.

[82] S. Dass, J. Yapeter, J. Zhang, J. Zhang, K. Pertsch, S. Nikolaidis, and J. J. Lim, "CLVR jaco play dataset," 2023. [Online]. Available: https://github.com/clvrai/clvr_jaco_play_dataset

[83] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, "Multi-stage cable routing through hierarchical imitation learning," *arXiv preprint arXiv:2307.08927*, 2023.

[84] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4126–4133, 2022.

[85] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," *CoRL*, 2023.

[86] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," 2021.

[87] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared Control Templates for Assistive Robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, p. 7.

[88] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[89] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[90] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *NeurIPS*, 2019.

[91] Y. Lin, A. S. Wang, G. Sutanto, A. Rai, and F. Meier, "Polymetis," https://facebookresearch.github.io/fairo/polymetis/, 2021.

[92] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "FMB: A functional manipulation benchmark for generalizable robotic learning," https://functional-manipulation-benchmark.github.io, 2023.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[94] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 205–11 212, 2022.

[95] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[96] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

Appendix

A. Related Work

Many works train policies using a large dataset of trajectories collected from a robot, from early efforts using autonomous data collection for scaling policy training [25], [26], [27], [28], [29], [30] to more recent efforts that explore the combination of modern transformer-based policies with large demonstration datasets [10], [31], [32], [33], [34], [35]. These works primarily focus on a single embodiment, while Octo trains policies on robot datasets assembled across *multiple* embodiments, increasing the effective size of the training dataset and allowing finetuning to a range of robot setups.

More recently, papers have focused on broadening the generalization abilities of robot policies. Multiple works leverage diverse non-robot data or pretrained vision-language foundation models to boost policy generalization to new scenes and tasks [35], [9], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [8]. More closely related to Octo are recent works that train robot policies across data from multiple robot embodiments: the GNM model [11], [17] generalizes across robot navigation setups while RoboCat [7] and RT-X [1] control multiple single-arm manipulation robots. While these models deliver impressive policy learning results, a key issue is their lack of flexibility: they typically require users to stick to the sensory inputs and action space used during pretraining and do not support adaptation to new observation and action spaces. Furthermore, the largest models are not publicly accessible. Octo differs from these works in multiple aspects: it is trained on a larger and more diverse robot data mix, it supports a wider range of downstream applications via efficient finetuning to new robot setups, and it is fully open source and reproducible.

Octo's design is inspired by several recent advances in robot imitation learning and scalable transformer training, including the use of denoising diffusion objectives [23] for action decoding [18], [47], [48], the prediction of "action chunks", i.e., sequences of future actions [32], [18], [33], and model layouts and learning rate schedules inspired by the literature on scalable vision transformer training [20], [49]. Our work is the first to leverage these approaches in the context of learning cross-embodied generalist policies and we find that they can lead to substantial performance improvements. In our evaluation, we present ablations to assess the importance of these components, alongside a more comprehensive list of what we found to be (un)important in Appendix E; we hope our findings are useful for future research on generalist policy learning.

A key ingredient for training generalist robot policies is robot training data. In contrast to vision and language data that can be scraped from the web, obtaining robot data at scale is challenging and often involves significant investments in hardware and human labor. There are multiple large robot navigation and autonomous driving datasets [50], [51], [52], [53], [17], [54], [55]. In recent years, there have also been multiple efforts for building robot *manipulation* datasets of increasing scale and diversity, either collected via scripted and autonomous policies [28], [27], [56], [57], [25], [30] or human teleoperation [58], [59], [60], [61], [62], [10], [63], [64], [65], [66], [67]. Octo is trained on the Open X-Embodiment dataset [1], a recent effort that pooled many of these aforementioned robot datasets. The Open-X dataset contains approximately 1.5M robot episodes, of which we curate 800k for Octo training. We note that the RT-X model [1] used a more restricted subset of 350K episodes, so to the best of our knowledge, Octo is trained on the largest robotics manipulation demonstration dataset to date.

B. Octo Code Example

Loading a pretrained Octo model and performing inference requires little code:

```python
import jax
from octo.model.octo_model import OctoModel

model = OctoModel.load_pretrained("checkpoint")
print(model.get_pretty_spec()) # Print out the
    input-output spec
observation = {"image_primary": img}
task = model.create_tasks(texts=["pick up the fork"
    ])
action = model.sample_actions(
    observation, task, rng=jax.random.PRNGKey(0))
```

Listing 1: Example Python code to perform inference with a pretrained Octo model.

C. Data mixture

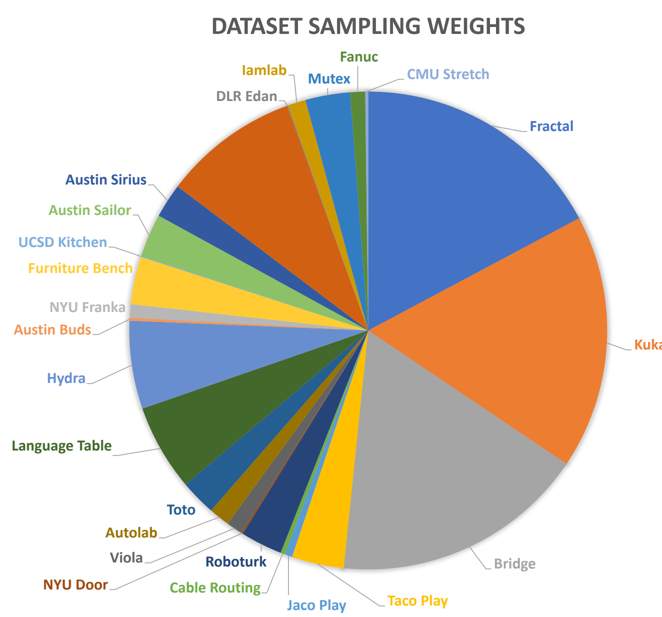

Fig. 6: **Training dataset composition.** We curate a subset of 25 datasets from the Open X-Embodiment dataset that have image observations, end-effector actions, and show diverse behaviors. The pie chart visualizes the fractions that each dataset contributes to every training batch on average. The dataset weights are determined by the number of samples in each dataset with small modifications to balance dataset size and diversity (see Section II-B for details).

We list the detailed training mixture used for training the Octo models in Table II. The sampling weights are mostly determined by the relative size of the datasets with a few manual adjustments (see Section II-B).

Octo Pretraining Dataset Mixture	
Fractal [10]	17.0%
Kuka [27]	17.0%
Bridge[60], [61]	17.0%
BC-Z [62]	9.1%
Stanford Hydra Dataset [68]	6.0%
Language Table [69]	5.9%
Taco Play [65], [66]	3.6%
Furniture Bench Dataset [70]	3.3%
UTAustin Mutex [71]	3.0%
Austin Sailor Dataset [72]	2.9%
Roboturk [73]	2.8%
Toto [74]	2.4%
Austin Sirius Dataset [75]	2.3%
Berkeley Autolab UR5 [76]	1.5%
IAMLab CMU Pickup Insert [77]	1.2%
Viola [78]	1.2%
Berkeley Fanuc Manipulation [79]	1.0%
NYU Franka Play Dataset [80]	0.9%
UCSD Kitchen Dataset [81]	<0.1%
Jaco Play [82]	0.6%
Berkeley Cable Routing [83]	0.3%
Austin Buds Dataset [84]	0.3%
CMU Stretch [85]	0.2%
NYU Door Opening [86]	0.1%
DLR EDAN Shared Control [87]	0.1%

TABLE II: Octo pretraining data mixture using datasets from the Open X-Embodiment dataset [1].

D. Training Hyperparameters

We mostly follow documented practices for training vision transformers [49]. We use the AdamW optimizer [88] with an inverse square root decay learning rate schedule [49] and linear learning rate warm-up. We list hyperparamaters used during training in Table III and the model parameters for the different sizes in Table IV. We apply standard image augmentations during training. Concretely, for the 3rd person camera we apply stochastic crops followed be a resize to 256×256, followed by color jitter. Finally, we normalize the input image to have pixels with float values between -1.0 and 1.0. For the wrist camera, we apply the same procedure except without the random crop and resizing to 128×128 instead.

Hyperparameter	Value
Learning Rate	3e-4
Warmup Steps	2000
LR Scheduler	reciprocal square-root
Weight Decay	0.1
Gradient Clip Threshold	1
Batch Size	2048

TABLE III: Hyperparameters used during training.

The images are passed through a shallow convolution stack, then split into a sequence of flattened patches [20] of size 16×16. This results in 256 tokens for the 3rd person camera images and 64 tokens for the wrist camera images. For datasets containing language annotations, we use a pretrained `t5-base` (111M) transformer model [19] that produces a sequence of 16 language embedding tokens.

Model	Layers	Hidden size D	MLP size	Heads	Params
Octo-Small	12	384	1536	6	27M
Octo-Base	12	768	3072	12	93M

TABLE IV: Architecture details of Octo model variants.

The diffusion action head consists of a 3-layer MLP with a hidden dimension of 256, residual connections, and layer normalization. We use the standard DDPM objective as introduced by [23] with a cosine noise schedule [89] and 20 diffusion steps.

E. Things that Worked and Did Not Work (Yet)

Things we found improved performance:

- **Adding history during pretraining**: Models with one frame of history as context performed better in zero-shot evals than models pretrained without history. We did not observe benefits of increasing the history length further on the few tasks we evaluated on, though other tasks may benefit.
- **Using action chunking**: We found it helpful to use "action chunking" [32], i.e., to predict multiple actions into the future, for getting more coherent policy movements. In our evaluations, we did not find temporal ensembling of future actions to provide additional benefits beyond receding horizon control.
- **Decreasing patch size** Tokenizing images into patches of size 16×16 led to improved performance over patches of size 32×32, particularly for grasping and other fine-grained tasks. This does add compute complexity (the number of tokens is $4\times$), so understanding how to balance compute costs and resolution remains a problem of interest.
- **Increasing shuffle buffer size**: Loading data from 25 datasets in parallel is a challenge. Specifically, we found that achieving good shuffling of frames during training was crucial — zero-shot performance with a small shuffle buffer (20k) and trajectory-level interleaving suffered significantly. We solved this issue by shuffling and interleaving frames from different trajectories *before* decoding the images, allowing us to fit a much larger shuffle buffer (up to 500k). We also subsample at most 100 randomly chosen steps from each training trajectory during data loading to avoid "over-crowding" the shuffle buffer with single, very long episodes.

Things that did not work (yet)::

- **MSE Action Heads**: Replacing our diffusion decoding head with a simple L2 loss led to "hedging" policies that move very slowly and e.g., fail to rotate the gripper in WidowX evaluations.
- **Discrete Action Heads**: Discretizing actions into 256 bins per dimension and training with cross-entropy loss

like in [10] led to more "decisive" policies, yet they lacked precision and often missed the grasp.
- **ResNet Encoders**: did not scale as well to larger datasets in our evaluations (see Table VI), though they did outperform our ViT architecture when training from scratch on a small dataset (around 100 demonstrations).
- **Pretrained Encoders**: ImageNet pretrained ResNet encoders did not provide benefit on zero-shot evals, though may be confounded with ResNet architectures underperforming as mentioned above.
- **Relative Gripper Action Representation**: When aligning the gripper action representations of the different datasets, we tried (A) absolute gripper actions, i.e., actions are +1 when the gripper is open and 0 if it is closed, and (B) relative gripper actions, i.e., gripper action is +1/0 only in the timestep when the gripper opens/closes and 0.5 otherwise. We found that the latter tends to open/close the gripper less often since most of the training data represents "do not change gripper" actions, leading to a slightly higher grasp success rate. At the same time, the relative representation led to less retrying behavior after a grasp failed, which was ultimately worse. Thus, we chose the absolute gripper action representation.
- **Adding Proprioceptive Inputs**: Policies trained with propioceptive observations seemed generally worse, potentially due to a strong correlation between states and future actions. We hypothesize this might be due to a causal confusion between the proprioceptive information and the target actions [90].
- **Finetuning Language Model**: In order to improve the visuo-lingual grounding of Octo we experimented with: i) varying sizes of the T5 encoder [19]: `small` (30M), `base` (111M), and `large` (386M) as well as ii) finetuning the last two layers of the encoder. Using the frozen `base` model resulted in the best language-conditioned policies. We did not find improvements when using larger encoders or finetuning the encoder. We hypothesize this might be due to the lack of rich, diverse, free-form language annotations in most of the datasets.

F. Experimental Setups

Zero-Shot Evaluations:

WidowX BridgeV2: Uses the setup of [61], in which a Trossen WidowX 250 6-DOF robot performs diverse manipulation tasks. The observation consists of a single third person camera stream and the action space is end-effector position deltas. We evaluated two language-conditioned tasks in which a the robot needs to "place the carrot on plate", and "put the eggplant in the pot." While these tasks are in-distribution, the policy must still generalize to novel object positions. We performed 10 trials per task and varied objects positions between trials.

UR5: Uses the setup of [76], in which a UR5 robot arm performs multiple table top manipulation tasks. The observation consists of a single third person camera stream and the action space is end-effector position deltas. We evaluated twp language-conditioned tasks: picking a toy tiger from a bowl and placing it into a different bowl as well as wiping a table with a cloth. While these tasks are in-distribution, the policy must still generalize to novel object positions, distractor objects, and lighting. Since the training data was collected months ago and the robot setup was taken down and re-assembled, the policy must also generalize to other miscellaneous changes in the environment like a slightly different camera view and background. We performed 10 trials per task and varied objects positions between trials.

RT-1 Robot: Uses the setup of [10], in which a proprietary robot performs multiple table top and furniture manipulation tasks. The observation consists of a single third person camera stream and the action space is end-effector position deltas. We evaluated on the task of picking up a 7up can, apple, blue chip bag, or brown chip bag, as well as the task of opening or closing drawers on a cabinet. While these tasks are in-distribution, the policy must still generalize to novel object positions. We performed 10 trials per task and varied objects positions between trials.

G. Model Ablations

All of our model ablations were evaluated on the WidowX setup. We present a more detailed breakdown of the success rates per task in Table V. We evaluated on two language-conditioned tasks (put carrot on plate and put eggplant in pot) and two goal-conditioned tasks (put bread on plate and put spoon on glove). The goal-conditioned tasks contain objects not seen in the Bridge dataset.

Finetuning Evaluations:

Lab3 Baking: The robot must pick up the toy bread object, place it in the toaster, and shut the toaster. This task requires generalization across initial positions (of both the toaster and object) and the shape of the target toy bread object. We use an end-effector delta action space (Cartesian position + rotation delta). Observations come from the 3rd-person front-facing Zed camera. Actions are predicted at 15 Hz, and executed on the robot using the R2D2 Franka controller. The finetuning dataset consists of 120 demos collected via expert VR tele-operation, and every policy was evaluated using 20 trials (4 novel test objects with 5 positions each).

Lab2 Coffee: The robot is tasked with picking up one of four different Keurig Coffee Pods and placing it inside of a Keurig machine. This task requires both generalization across initial positions and colors of the coffee pod, as well as precise placement in the Keurig machine. We use an end effector delta action space with an open source controller running at 10 Hz based on Polymetis [91] (found here). We use only a single 3rd-person wrist observation. Our training dataset contained 118 expert demonstrations from varied coffee pods and positions collected via VR tele-operation. We evaluated policies for 20 episodes, five episodes for each of four different color coffee pods.

Lab1 Peg Insertion: The task is to insert a pre-grasped 3D-printed peg into a matching slot on a 3D-printed board

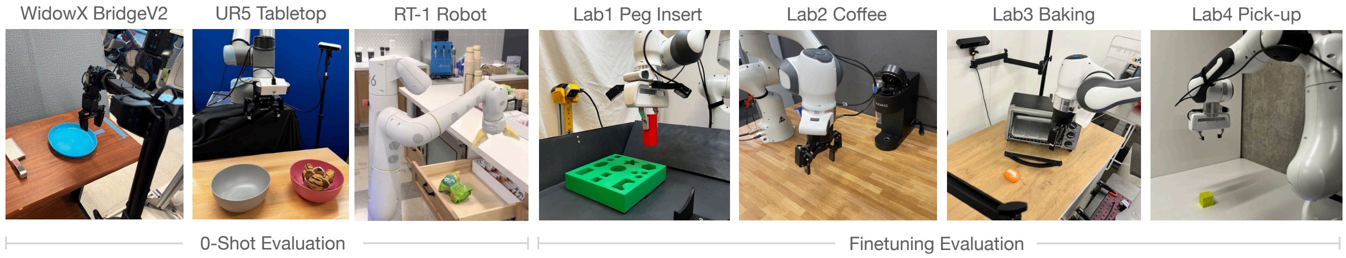

Fig. 7: **Evaluation Tasks.** Replicated from the main text for convenience. We evaluate Octo on 7 real robot setups across 4 institutions in zero-shot and finetuning scenarios.

		Put carrot on plate	Put eggplant in pot	Put bread on plate	Put spoon on glove	Average
	Octo-small (Ours)	80%	90%	70%	90%	**83%**
DATA	RT-X dataset mix [1]	80%	80%	40%	40%	60%
DATA	Single robot dataset (Bridge Data)	20%	70%	60%	20%	43%
POLICY	Discretized Action Prediction [1]	0%	20%	10%	40%	18%
POLICY	Continuous Action Prediction (MSE)	70%	30%	0%	40%	35%
ARCH	Resnet-50 + Transformer [1]	80%	60%	100%	40%	70%

TABLE V: **Model Ablations.** We achieve best performance when using the ViT architecture, diffusion action head, and wide training data mixture. All evaluations are performed on the WidowX setup. Success rates are averaged over 40 trials across two language-conditioned tasks and two goal-conditioned tasks.

inside the bin. The matching tolerance between the peg and the hole is 1.5mm; which makes it a contact-rich precise part-mating task. The robot must learn an appropriate policy to “search" for the matching opening through contact, which necessitates the use of force/torque measurements. The observation space of the policy consists of a single side-view camera image, the end-effector twist, and the end-effector force/torque reading. The policy sends action commands as the robot's end-effector twists at 5 HZ, tracked at 1000 HZ by a low-level impedance controller. Our finetuning dataset is composed of 100 human demonstrations from the FMB dataset [92]. We evaluated trained policies for 20 trials with randomized board positions.

Lab4 Pick Up: We use the setup of [22]. The robot needs to pick up a block from a table top surface after being trained on a dataset of 100 pickups of various objects. The robot uses joint position control with an underlying Polymetis control stack [91] (here). It is conditioned on a wrist camera input image as well as the proprioceptive readings of the robot. We evaluted on the task of picking up the yellow cube and used 20 trials.

Design Decisions for Generalist Robot Policy Training: We have demonstrated the effectiveness of Octo as a zero-shot multi-robot controller and as an initialization for policy finetuning. We next analyze the effects of different design decisions on the performance of the Octo policy. Concretely, we focus on the following aspects: (1) model architecture, (2) training data, (3) training objective, and (4) model scale. Unless noted otherwise, we perform all ablations on the Octo-Small model due to our compute budget.

		Aggregate Performance
	Octo-Small (Ours)	**83%**
DATA	RT-X dataset mix [1]	60%
DATA	Single robot dataset (Bridge Data)	43%
POLICY	Discretized Action Prediction [1]	18%
POLICY	Continuous Action Prediction (MSE)	35%
ARCH	Resnet-50 + Transformer[1]	70%

TABLE VI: **Model Ablations.** We achieve best performance when using the ViT architecture, diffusion action head, and wide training data mixture. All evaluations are performed on the WidowX setup. Success rates are averaged over 40 trials across two language-conditioned tasks and two goal-conditioned tasks.

Model architecture: Prior transformer-based policy designs typically encode input images with large ResNet-style [93] encoders and fuse the resulting image features with a comparatively small transformer [10], [1], [11], [18], [32], [94], [34]. Instead, we opt for a “transformer-first” architecture that uses very shallow CNN patch encoders and concentrates most of the parameters and FLOPS in the transformer backbone, similar to canonical vision transformer architectures [20]. In Table VI we show that this scalable architecture leads to substantially improved performance when training on the full Open X-Embodiment data mix. Importantly, we found ResNet-based architectures to

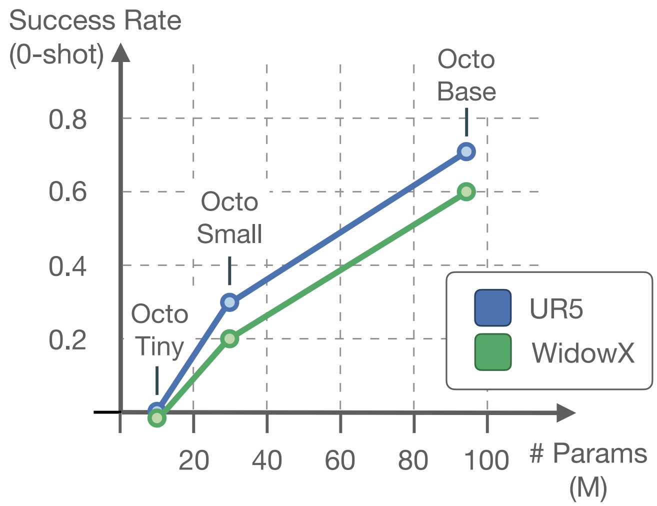

Fig. 8: **Model Scaling.** The performance of Octo improves with larger model sizes on both UR5 and WidowX tasks. Success rates are averaged over 10 trials on one language-conditioned task per robot.

perform better than ViTs when training on small datasets, e.g., in our "from scratch" comparisons, underlining that large transformer policies are uniquely suited for scalable training on diverse datasets.

Training data: Octo is trained on the most diverse cross-embodied robot dataset to date, a mix of 25 datasets that we manually curated from the Open X-Embodiment dataset [1] (see Section II-B). We ablate the impact of this training mix by comparing to Octo models trained on a smaller mix of 11 datasets used in training the RT-X models [1] and a baseline trained only on data from the target robot domain. In Table VI we show that the performance of Octo increases as we increase the number of training datasets. This suggests that expanding the data mix to even more datasets may further improve policy performance. We will leave this for future work, along with a more thorough investigation of best practices for data curation.

Training objective: We compare Octo's diffusion decoding training objective (see Section II-C) to common alternatives from prior work: simple MSE loss [95], [96] and cross-entropy loss on discretized actions [10], [9]. In Table VI we find that Octo's diffusion training objective leads to substantially improved performance. This improvement is likely because the diffusion head can model multi-modal action distributions (unlike the MSE head) while retaining the precision of continuous actions (unlike the discrete head). Qualitatively, the policy acts more decisively than MSE-trained policies, and more precisely than those trained with discretized actions.

Model scale: We compare Octo models of three different sizes following the ladder of common vision transformer models [49]: Octo-Tiny (10M), Octo-Small (27M), and Octo-Base (93M). In Figure 8 we show that the zero-shot performance of the policy scales with increasing model size. We find that the Base model is more robust to initial scene configuration than the Small model, and is less prone to early grasp attempts, indicating the larger model has better visual scene perception.