# HEETR: Pretraining for Robotic Manipulation on Heteromodal Data

**Garrett Thomas** [1]
gwthomas@stanford.edu

**Andrey Kolobov** [2]
akolobov@microsoft.com

**Ching-An Cheng** [2]
chingan.cheng@microsoft.com

**Vibhav Vineet** [2]
vibhav.vineet@microsoft.com

**Mihai Jalobeanu** [2]
mihaijal@microsoft.com

[1] Department of Computer Science, Stanford University
[2] Microsoft Research

**Abstract:** A good representation is a key to unlock efficient learning for real-world robot manipulation. However, common manipulation-relevant datasets do not always have all the modalities (e.g., videos, actions, proprioceptive states) presented in robotic manipulation. As a result, existing approaches to representation learning, which assume full data modalities, cannot be easily scaled to consume all the data; instead, they can only be applied to a subset of modality sufficient data, which limits the effectiveness of representation learning. In this work, we present an end-to-end transformer-based pretraining method called *HEETR* (**H**eteromodal **E**nd-to-**E**nd **T**ransformer for **R**obotic manipulation) that can learn a representation for efficient adaptation using all data regardless of their available modalities. We demonstrate the merits of this design and establish new state-of-the-art performance on Robosuite/Robomimic and Meta-World benchmarks.

## 1 Introduction

Recent progress in training large-scale general-purpose representations, sometimes called *foundation models (FM)* [1], has significantly advanced computer vision (CV) and natural language processing (NLP). But it hasn't had nearly the same effect on robotic manipulation. While many attribute this to the lack of sufficient quantities of robotics-relevant data, we argue that this field's data issue is more subtle and propose a new method that exploits the structure of available robotic manipulation data for effective representation learning.

Ideal training data for robotic manipulation is diverse and multimodal, consisting of *matching* sequences of video frames, depth maps, proprioceptive states, control inputs, rewards, language instructions, etc. However, realistically available manipulation-relevant datasets are *heteromodal*: each of them typically has only a subset of these modalities. Some of these datasets – those comprised of annotated videos – are actually very large [2–5]. Unfortunately, since they don't include control inputs, they aren't enough per se to learn a *percepto-motor* representation that could drive a robot directly. We call such datasets *modally deficient (MD)* in constrast to the ideal *modally sufficient (MS)* datasets containing both percepts and actions. MD datasets are typically used to train *perceptual* representations [6–8], which are then adapted with the help of MS data at test time.

Because MS datasets are more expensive to collect, existing MS datasets [9, 10] are not sufficiently large to learn a percepto-motor representation as general as, e.g., GPT-3 for NLP [11]. In the meantime, findings from CV literature [12] suggest that pretraining with a small amount of labeled mul-

CoRL 2022 Workshop on Pre-training Robot Learning, Auckland, New Zealand.

titask (MS) data combined with large amounts of unlabeled data of a single modality (MD) can significantly reduce adaptation-time data needed for achieving good target task performance. The question is, then: *can we learn good representations by mixing abundant MD and scarce MS data in an end-to-end pretraining process for robotic manipulation as well?*

In this work, we give a preliminary affirmative answer to this question. We present an end-to-end transformer-based pretraining method called *HEETR* (**H**eteromodal **E**nd-to-**E**nd **T**ransformer for **R**obotic manipulation) that leverages MD and MS data in concert for efficient adaptation. Many works have applied transformers to model sequential decision-making [13–20] by autoregressive prediction-based losses. These losses work well on MS data, which contain actions that serve as stable prediction targets. However, MD datasets with only high-dimensional percepts (e.g., videos) present a problem. Predicting raw video frames autoregressively in pixel space is very expensive and captures unnecessary detail, while autoregressive prediction in a latent space can easily lead to representation collapse. Other than freezing perceptual modules, one can also use contrastive or reconstructive losses [21, 22], but they depend on tricky-to-tune aspects such as the choice of negative examples.

Fortunately, representation collapse is not an issue for autoregressive latent prediction on MD datasets as long as *(a)* the training process leverages MD and MS data *at the same time*, and *(b)* the two kinds of datasets share some of the high-dimensional modalities. We design HEETR to alternate between sampling batches from MD and MS datasets. In this way, the action prediction loss defined on the MS data can act as a representation stabilizer while the autoregressive latent prediction loss defined on the much larger MD data can enable extracting a rich set of features. We demonstrate the merits of this design by establishing new state-of-the-art performanace across 9 challenging Robosuite/Robomimic [23, 24] tasks, and on the most challenging split of the Metaworld benchmark [25], ML50.

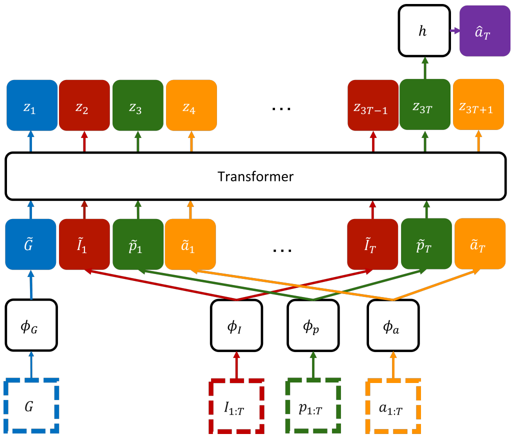

Figure 1: HEETR architecture. Each input is passed through a modality-specific encoder, if present; a trainable embedding is used for missing inputs. The encoded sequence is processed by a transformer and used to predict the next action.

## 2 HEETR architecture and training

**Architecture.** The architecture design is depicted in Figure 1. We base our model on Decision Transformer (DT) [14] (which is based on GPT-2 [26]), but make several important changes:

- **Positional embedding for variable trajectory length.** The length of trajectories varies widely across realistic robotic manipulation datasets. Therefore, unlike DT, HEETR doesn't impose a positional embedding based on the maximum trajectory horizon but uses a relative positional embedding. HEETR assigns positional embeddings based on an entry's position in a fixed-size window as opposed to its time step used by DT. This relative embedding scheme makes HEETR more data efficient than the original DT version when the trajectory lengths are not fixed.
- **No rewards.** Unlike DT, HEETR doesn't use rewards or returns and operates in a (multitask) imitation learning mode. Realistic robotics data come at best with sparse rewards that are heuristically assigned. We found pretraining a model to condition on returns of such rewards to be problematic, because of variability in trajectory lengths across tasks and even across initial states within a task. We believe more research is needed to allow robust learning from suboptimal trajectories in common multitask, reward-impoverished robotic manipulation scenarios.
- **Goal conditioning.** To enable HEETR to generate a trajectory for a given task, we train it to condition on task descriptions in the form of goal images. Conditioning on other ways of specifying tasks such as language can be easily used instead [17, 27, 28].

- **Input modalities, embeddings, and augmentations** In addition to goal images, HEETR inputs proprioceptive states, multicamera image observations, and actions as inputs at each time step. It can be easily extended to support others, such as depth maps and language instructions. Actions and proprioceptive states are encoded using linear layers. For multicamera image observations, we adopt the encoder from [24] (which applies a random crop augmentation to each image, passes it through a ResNet18 instance, then a spatial softmax layer [29], and finally a small MLP).
- **Missing-modality placeholders.** For each modality, HEETR has a learnable vector used in place of the encoder when that modality is missing from a given trajectory/dataset.

The embeddings are passed to the transformer according to Fig. 1 and finally actions are predicted by a linear head applied to the transformer output.

**Pretraining.** We make the following assumptions on data:

1. HEETR has access to a collection $\mathscr{D}$ of datasets. Each dataset $\mathcal{D} \in \mathscr{D}$ consists of sequences over a set of modalities $\mathscr{M}_{\mathcal{D}}$ and has a set of losses $\mathscr{L}_{\mathcal{D}}$ defined over modalities in $\mathscr{M}_{\mathcal{D}}$.
2. Each dataset $\mathcal{D} \in \mathscr{D}$ shares at least one modality $\mathcal{M}$ with at least one other dataset $\mathcal{D}' \in \mathscr{D}$, and at least one loss associated with each dataset uses modality $\mathcal{M}$ as input. Note that this does *not* imply that $\mathcal{D}$ and $\mathcal{D}'$ have to have any data in common or share all modalities.
3. At least one dataset $\mathcal{D} \in \mathscr{D}$ is MS (i.e., contains both actions and perceptions), and at least one loss in $\mathscr{L}_{\mathcal{D}}$ uses actions as the prediction target.

For example, suppose we have MS $\mathcal{D} = Bridge\ dataset$ [10] and MD $\mathcal{D}' = Ego4D$ [2]. They have the video modality in common. If we define a video frame embedding prediction loss over $\mathcal{D}'$ and an action prediction loss over $\mathcal{D}$ that uses video frames as input, then sets $\mathcal{D}$ and $\mathcal{D}'$ satisfy the assumptions above. However, note that videos contained in each dataset are completely distinct.

We associate a probability $p_{\mathcal{D}}$ to each dataset such that $\sum_{\mathcal{D} \in \mathscr{D}} p_{\mathcal{D}} = 1$ and a coefficient $\lambda_{\mathcal{L}} \geq 0$ with each loss $\mathcal{L} \in \bigcup_{\mathcal{D} \in \mathscr{D}} \mathscr{L}_{\mathcal{D}}$. HEETR pretraining consists of repeatedly sampling batches from different datasets $\mathcal{D}$ with probability $p_{\mathcal{D}}$ and minimizing $\mathcal{L}_{\text{pre}}(\theta) = \mathbb{E}_{\mathcal{D} \sim p_{\mathcal{D}}} \sum_{\mathcal{L} \in \mathscr{L}_{\mathcal{D}}} \lambda_{\mathcal{L}} \mathcal{L}(\theta)$. The losses used in this work are described below:

- For $\tau$ with actions, an action prediction mean squared error (MSE) loss $\mathcal{L}_A(\tau; \theta) = \sum_t \|a_t - \hat{a}_\theta(\tau)_t\|_2^2$, where $a_t$ is the action at time $t$, $\hat{a}_\theta(\tau)$ is the HEETR's predictions for all actions in the sequence. Each $\hat{a}_\theta(\tau)_t$ depends only on the given goal and history up to time $t$ within the window.
- A video prediction loss that encourages the transformer representation to contain information useful for predicting the future. For each $t \in [1, T - k]$, HEETR minimizes a MSE loss $\mathcal{L}_V^{(k)}$ between a prediction of the image embedding at time $t + k$ using information available up to time $t$, and the output of the encoder. Note that if a dataset contains only videos, $\mathcal{L}_V^{(k)}$ is autoregressive and can lead to representation collapse if not stabilized by another loss such as $\mathcal{L}_A$.

**Fine-Tuning.** At test time, we are given a dataset $\mathcal{D}_{\text{ft}}$ containing MS demonstrations of a single target task. We update the model using behavioral cloning, minimizing $\mathcal{L}_{\text{ft}}(\theta) = \mathbb{E}_{\tau \sim \mathcal{D}_{\text{ft}}}[\mathcal{L}_A(\tau; \theta)]$. To mitigate overfitting, some of the model's parameters may be frozen during this stage,

## 3 Experiments

We first validate HEETR in a single-task, no-pretraining mode on 9 Robosuite/Robomimic tasks, and then demonstrate its utility in a multitask pretraining mode on Metaworld.

**Robosuite/Robomimic: HEETR's performance in a challenging single-task mode.** To validate HEETR's architecture, we train it in single-task behavior cloning (BC) mode on 9 Robosuite's single-arm (Panda, 6 DoF) tasks. For each task, we gather 75 demonstrations by controlling the arm using Robosuite's keyboard interface. HEETR achieves the following success rates across 5

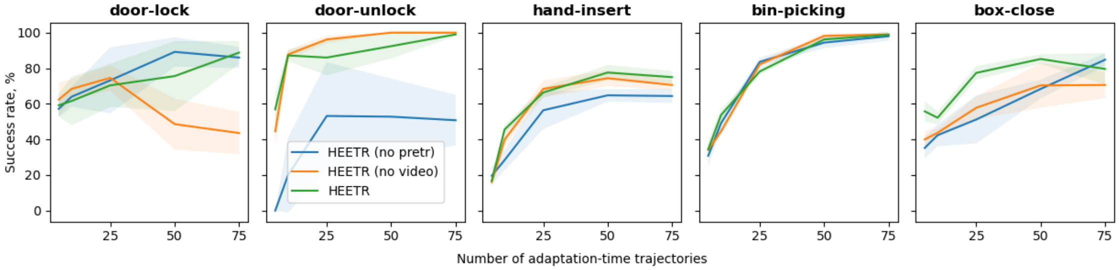

Figure 2: Meta-World results

seeds: Lift – 97.6%, Stack – 84%, Door – 67.6%, PickPlaceCan – 77%, PickPlaceMilk – 54%, PickPlaceBread – 76.4%, PickPlaceCereal – 69.6%, NutAssemblySquare – 55.6%, NutAssembly-Round – 46.4%. To our knowledge, this is state of the art on these tasks for learning with sparse or no rewards on this amount of data. Our main point of comparison is BC-RNN [24]'s results, but they are available only for Robomimic's NutAssemblySquare, PickPlaceCan, and Lift data. On the data we gathered for other tasks, that BC-RNN implementation achieves the success rate of 0.

**Meta-World: evaluating HEETR in the pretraining mode.** Recall that HEETR was designed to take advantage of pretraining jointly on MD and MS datasets. In this section, we compare HEETR's performance in the single-task behavior cloning mode to the performance of HEETR pretrained on video-only as well as on a MS multitask dataset and then adapted to individual tasks.

Specifically, we consider the ML50 split of Meta-World [25], which consists of 45 training tasks and 5 target tasks: door-lock, door-unlock, hand-insert, bin-picking, and box-close. We use the version of Meta-World with high-dimensional observations: at each time step, the agent receives an image from the task's *corner* camera and the Sawyer arm's 18-dimensional proprioceptive state. For each of the 5 target tasks, we use Meta-World's provided scripted policies with noise set to 0 to generate 75 demonstration trajectories. We then produce several datasets for each task: containing 5, 10, 25, 50, and 75 shortest trajectories out of the 75. As a baseline, we train HEETR in the single-task mode on each of these datasets for each task, and average the success rate of the resulting policies over 10 seeds. For a given task, we train for 10 epochs of 500 sampled batches, each batch consisting of 256 trajectory segments of length 30, at the learning rate of $5 \times 10^{-4}$. The resulting performance is presented in the **HEETR (no pretr)** plots in Figure 2.

To measure the effect of HEETR pretraining, we generate two pretraining datasets using the remaining 45 tasks. For 3 of these tasks – pick-out-of-hole, door-open, and pick-place-wall – we generate 15 trajectories per task using the aforementioned scripted policies. These trajectories comprise our MS pretraining dataset. For the remaining 42 tasks, we generate 100 trajectories per task using the aforementioned scripted policies, but record only the *video frames*. These 4200 videos form our MD dataset. For comparison, we perform pretraining with (**HEETR** plots in in Figure 2) and without using the MD dataset (**HEETR (no video)** plots in in Figure 2). During pretraining for **HEETR**, we sample batches from the MD dataset with $p = 0.3$ and apply the video frame embedding prediction loss to them. The batches from the MS dataset are sampled with $p = 0.7$ and are used to compute both the video loss and the action prediction loss. After pretraining for 15 epochs, we adapt the resulting HEETR model to each of the 5 target tasks by continuing to train it for 10 epochs on the 5-, 10-, 25-, 50-, and 75-trajectory per-task datasets mentioned earlier.

There are two notable trends in the results. (1) **HEETR** matches or outperforms the other models for most tasks and adaptation dataset sizes. (2) The MS dataset's tasks are reasonably similar to the door-unlock, hand-insert, and bin-picking target tasks. **HEETR (no video)** does well on these tasks and matches **HEETR**'s performance on them. But on door-lock and box-close, **HEETR** performs better. We conjecture that this is due to **HEETR** seeing more diverse video data during pretraining.

**Acknowledgments**

We would like to thank the reviewers for their time.

## References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models, 2021.

[2] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2022.

[3] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130: 33–55, 2022.

[4] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[5] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016.

[6] A. S. Chen, S. Nair, and C. Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos. In *RSS*, 2021.

[7] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation, 2022.

[8] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real world robot learning with masked visual pre-training. In *CoRL*, 2022.

[9] 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 *CoRL*, 2019.

[10] 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. In *RSS*, 2022.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[12] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[13] S. Dasari and A. Gupta. Transformers for one-shot visual imitation. In *CoRL*, 2020.

[14] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*, 2021.

[15] M. Janner, Q. Li, and S. Levine. Reinforcement learning as one big sequence modeling problem, 2021.

[16] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022.

[17] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.

[18] H. Kim, Y. Ohmura, and Y. Kuniyoshi. Transformer-based deep imitation learning for dual-arm robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8965–8972. IEEE, 2021.

[19] A. Prakash, K. Chitta, and A. Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021.

[20] C. R. Dance, J. Perez, and T. Cachet. Conditioned reinforcement learning for few-shot imitation. In *International Conference on Machine Learning*, pages 2376–2387. PMLR, 2021.

[21] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021.

[22] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

[23] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning, 2020.

[24] 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. In *CoRL*, 2021.

[25] T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2019.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

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

[28] 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*, pages 991–1002. PMLR, 2022.

[29] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.