An Unbiased Look at Datasets for Visuo-Motor Pre-Training

Anonymous Author(s)
Affiliation
Address
email

Abstract: Visual representation learning is a key challenge in robotics, but is severely hampered by the scarcity and homogeneity of robotics datasets. Recent works address this problem by pre-training large-scale representations on out-of-domain data (e.g., videos of egocentric interactions) and then *transferring* them to target robotics tasks. While the field is heavily focused on developing better pre-training algorithms, we find that dataset choice is just as important to this paradigm's success. After all, the representation can only learn the structures or priors present in the pre-training dataset. To this end, we flip the focus on algorithms, and instead conduct a *dataset centric* analysis of robotic pre-training. Our findings call into question some common wisdom in the field. We show that standard vision datasets (like ImageNet and Kinetics) are surprisingly competitive options for visuo-motor representation learning, and that the pre-training dataset's diversity matters more than its sheer size. Finally, we find that common simulation benchmarks give a misleading sense of progress, and that simple regularization strategies can dramatically improve real world policy learning.

Keywords: Visual Representation Learning, Datasets, Manipulation

1 Introduction

Consider a robot tasked to perform a manipulation task in an unstructured environment: e.g. toasting a bread slice. To accomplish this, the robot must locate the target objects (bread, toaster, etc.) in the scene and reason about their physical properties (e.g., Center-of-Mass, friction, etc.), using RGB camera inputs. However, the real world has innumerable objects, lightning conditions, and environments that a robot may run into. This incredible diversity makes hand-engineering a vision pipeline nearly impossible. Thankfully, the computer vision and representation learning communities have highlighted a successful paradigm to overcome this challenge: they automatically learn end-to-end neural representations *directly from data* [2, 3], which can then be used for downstream vision tasks. We seek to do the same for policy learning.

But what data should these representations be trained on? In an ideal world, we could simply leverage task-specific robotic data (i.e. trajectories) to jointly learn a representation and a controller, using end-to-end policy learning algorithms (e.g. Reinforcement Learning [4, 5], Behavior Cloning [6, 7], etc.). Unfortunately, learning visual representations in this manner requires a large amount of data that is too expensive to collect on real hardware. Furthermore, the lack of diversity in robotics data (e.g. collected

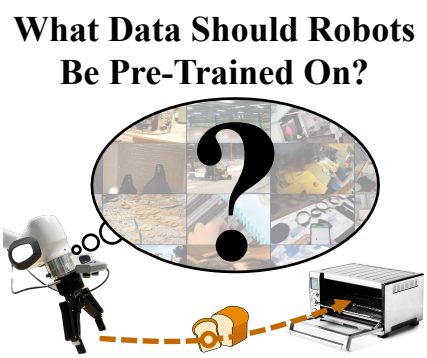

Figure 1: Due to the scarcity of diverse, large-scale robotic data, visuo-motor representations – which are necessary to solve tasks (e.g. put bread in toaster) from visual inputs – must be learned from external datasets [1]. But which datasets contain the best priors for robotics? We address this question by analyzing 5 diverse datasets from a robot learning perspective. Surprisingly, we find that standard vision datasets (e.g. ImageNet) can outperform SOTA robotics baselines, despite training on ~ 5x less data.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

in single lab) hinders generalization to diverse scenarios, which is the motivation for learning in the first place! To overcome this, the field has trended towards *pre-training* visual representations on large-scale, unlabeled vision datasets, using self-supervised learning algorithms – e.g. Masked Auto-Encoders [8] (MAEs) and contrastive learning [9, 10, 11], etc. These representations can effectively warm-start policy learning, thus allowing us to learn behaviors with far less robotic data [1, 12, 13].

The key insight is that self-supervised visual pre-training can learn useful robotics priors from *out-of-domain* data. Recent work [12, 1, 14, 13] has investigated different neural architectures and algorithms for learning these priors. However, there is one important commonality – all these methods train (primarily) on the same dataset, Ego4D [15]. Indeed, this seems like an intuitive choice, because: (1) Ego4D contains first-person camera views, which are analogous to the robot's camera; (2) Ego4D focuses on human-object interaction, i.e., it is aligned with the downstream manipulation task; and (3) the dataset offers a large number of images to train on. But are these *human intuitions* empirically useful for deciding what data to pre-train on? And is Ego4D really the best option?

In this paper, we investigate these research questions from the perspective of robotic manipulation tasks. Specifically, we pre-train a total of 15 representations on various datasets using MAEs [8], a state-of-the-art (SOTA) self-supervised learning algorithm. We then finetune each of these representations to solve diverse manipulation tasks in simulated and real settings via Behavior Cloning (w/ ≤ 50 demonstrations). These experiments clearly show that human intuition does **NOT** capture the complex attributes (e.g. dataset diversity) that are critical for learning generalizable representations. Surprisingly, we find that standard image datasets (e.g., ImageNet [2], Kinetics [16], 100 Days of Hands [17]) can all learn stronger visuo-motor representations than Ego4D! In fact, simply pre-training on ImageNet compares favorably against SOTA (visuo-motor) baseline representations, which were trained on far more data (e.g. MVP [18] was trained on 2M+ frames) using the exact same algorithm and hyperparameters. This leads us to an importance insight – the pre-training image distribution is far more critical for effective representation learning than the sheer number of frames trained on. Thus, we investigate various schemes for boosting batch diversity during pre-training. Our best model improves performance by 30% (v.s. SOTA baselines [13, 18]) on real world robotics tasks and is the direct result of this search. Finally, we show how simple implementation details (like using Dropout [19] during evaluation) can have a significant impact on policy performance, and how these trends are poorly captured in prior simulation studies. Our project code and models will be released publicly, and we encourage the reader to view our supplemental video for added context[1].

2 Related Works

Learning Actionable Representations The robotics field has long focused on learning actionable representations, which focus on task relevant details and are maximally predictive of the actions the robot should take. These representations can be learned end-to-end as part of policy learning, using data collected by expert demonstrations (e.g., Imitation Learning [20]) or the robot itself (e.g., Reinforcement Learning [4]). This paradigm has been successfully applied to a wide range of tasks like in-the-wild grasping [21], bin-picking [22, 23, 24], insertion [5], pick-place [25], and even self-driving [26, 27]. Prior work also added tertiary optimization objectives (e.g. observation reconstruction [28], inverse modeling [29], dynamics modeling [30], etc.) on top of policy learning, in order to make representation learning more efficient. However, all of these techniques share the same flaw: they require a wealth of task-specific robotic data for learning representations.

Self-Supervised Visual Pre-Training Thus, the robotics community has trended towards *pre-training* representations on out-of-domain, vision datasets, (which are both larger and more diverse) and transferring them to robotics tasks. Prior works [1, 14, 18, 12, 13] all seem to follow a common formula: representations are trained using SOTA self-supervised vision algorithms (e.g., contrastive learning [9, 11, 10], generative modeling [8], etc.) on frames (primarily) sampled from the Ego4D dataset [15]. These representations are then evaluated mostly in sim [31], using a common policy learning framework [1, 13]. While these choices seem reasonable (e.g., Ego4D has $400M+$ in-the-wild video frames, and sim is easy to share/reproduce), our investigation shows that it is critical to probe these assumptions. In fact, we show that standard image datasets are surprisingly applicable to the robotics space, and that common evaluation techniques can give a misleading sense of progress.

[1] https://sites.google.com/view/robotics-datasets-analysis

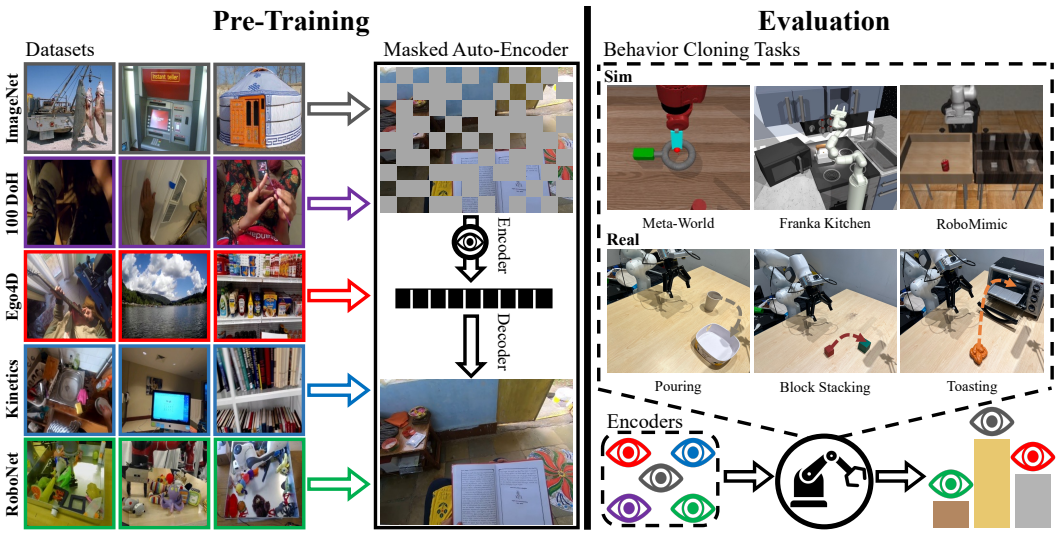

Figure 2: Our investigation considers 5 standard datasets from the both the vision and robotics fields: ImageNet [2], 100 Days of Hands [17] (DoH), Ego4D [15], Kinetics [16], and RoboNet [32] (left). For each dataset, we pre-train a visual representation on it using the Masked Auto-Encoders (MAE) algorithm [8]. This SOTA method works by randomly masking images, and training an encoder/decoder to reconstruct them (center). Once pre-training is concluded, we fine-tune the representation to various robotics tasks, both in simulation and in the real world (right). Performance on those tasks is used to quantify the representation's use-ability for robotics.

3 Experimental Methods

Our investigation follows a simple experimental formula (see Fig. 2): first we pre-train representations on various datasets using a (fixed) self-supervised algorithm, and then we fine-tune each representation to solve various manipulation tasks (both simulated and real) via behavior cloning. The representations were rated based on how successfully they completed the tasks, with an extra emphasis placed on the real world numbers. We adopted this framework because it closely tracks prior work (see Sec 2), and allows us to create more reproducible numbers. The following sections discuss each of these components in detail.

Visual Pre-Training This project requires a scalable representation learning algorithm that can seamlessly operate on heterogeneous data sources, with the highest possible performance. We chose to use Masked Auto-Encoders (MAEs), a self-supervised representation learning algorithm with SOTA performance on various vision [8] and robotics tasks [13, 18]. The MAE encoder (E) is a Vision Transformer (ViT) network [33] that produces an embedding vector to represent an input image I: i.e. $E(I) \in \mathcal{R}^{768}$. During training E is forced to represent I, using only *random patches* sampled from the image. Then a decoder network (D) attempts to reconstruct I in its entirety (see Fig. 1, middle). Both E and D are trained end-to-end using a MSE reconstruction objective: $\min_{D,E} ||D(E(I)) - I||_2$. During training E is forced to pay close attention to how patches relate to each other, and how they are joined to form the final image. Thus, it learns a highly efficient image descriptor that can be transferred to downstream tasks without any algorithmic changes (e.g. no masking needed during transfer). MAE hyperparameters are described in Appendix A. Note that they are directly copied from the original MAE paper [8] and shared by prior robotics works [13, 18].

Fine-Tuning w/ Behavior Cloning Once learned, the MAE representations are fine-tuned to solve manipulation tasks. To accomplish this we adopt the Learning from Demonstration [34, 35, 20] problem setting. Our goal is to learn a policy π that uses the given observation o_t to predict an optimal action distribution for the task: $a_t \sim \pi(\cdot|o_t)$. Note that the actions a_t are commands sent to the robot controller, while the observations consist of the current image and robot joint information: $o_t = [i_t, j_t]$. Policy π must be learned given a set of expert demonstrations ($\mathcal{D} = \{\tau_1, \ldots, \tau_n\}$),

where each demonstration $\tau_i = [(o_0, a_0), \ldots, (o_T, a_T)]$ is a trajectory with optimal observation-action tuples (i.e. collected by a proficient agent).

The policy π is parameterized using a 2-layer network (p), built atop the pre-trained encoder E. The forward pass works as follows: first, the observation image is encoded $E(i_t)$; then j_t is concatenated to the encoding and passed to the policy network – $p(E(i_t), j_t)$. p predicts a policy distribution, which in our case is a Gaussian Mixture Model [36, 37]. During test time, actions are sampled from this distribution and then executed on the robot. The entire policy network (both p and E) is fine-tuned end-to-end (using Adam [38] for $50K$ iterations) to maximize the log probability of actions from the expert demonstrations: $\min_\pi -log(\pi(a_t|E(i_t), j_t))$. This procedure was designed to closely match prior work with two important modifications: we apply dropout to the policy network p, and we apply data augmentation to i_t before passing it to the encoder. Both of these deviations are validated in our experiments (see Sec. 5). Please refer to Appendix B for the exact hyperparameters.

Evaluation Procedure To evaluate a representation, we apply our fine-tuning stack (see above) separately on 16 different simulated and real manipulation tasks. To get the final numbers, each policy is evaluated on N test rollouts, with novel initializations (e.g., test objects, new initial positions, etc.). We report the success rates (i.e. percentage of N rollouts that are completed successfully) for each policy at the end of training. Note that all evaluation hyperparameters (e.g., demonstration set, number of test-time rollouts, initial positions, test objects, BC hyperparameters, etc.) are kept fixed within a task. This ensures that numbers for each representation are directly comparable against each other, and that noise during the testing procedure is minimized.

We now describe the simulated (13 tasks) and real (3 tasks) evaluations in detail (see Fig. 2, right).
Simulation: Our simulated tasks set spans a set of 3 MujoCo [31] environments. Specifically, we evaluate on 5 tasks from *Metaworld* [39] (BinPick, ButtonPress, Hammering, Drawer Opening, and Assembly), 5 tasks from *Franka Kitchen* [40] (Knob Turning, Door Opening, Light Switch, Microwave, and Sliding Door), and 3 tasks from *RoboSuite* [41, 36] (Lift, Can, and Square). These environments are frequently used by the robot learning community, and the exact setups (e.g., camera positioning, object sets, demonstration trajectories, etc.) were directly taken from prior work [13, 1, 36]. As a result, our simulated results should be very accessible to the community.
Real World: While simulation is a useful tool, there is a significant sim2real gap in manipulation. Thus, we designed 3 distinct tasks for real world validation on a Franka Panda Robot. (1) *Block Stacking* requires the robot to pick up the red block and place it on the green block. This is the simplest task, since the robot only has to adapt to new object configurations during test time, but it still requires the robot to precisely localize and grasp the (small) red block. (2) *Pouring* requires the robot to lift the cup and pour almonds in the target bowl. During test time the cup and target bowls are both novel objects (unseen during training), and are placed in random locations. Thus, this task forces the robot to generalize to new visual inputs. (3) *Toasting* is the final task, and it requires the robot to pick up the object, place it in the toaster, and then shut the toaster. During test time, we use a novel object and randomize both the object's initial pose and the toaster's initial orientation. This is the most difficult task, since it requires the robot to execute a multi-stage manipulation strategy, while also generalizing to new visual scenarios. We collect $n = 50$ demonstrations via VR teleop for each of the tasks, and present all other task parameters in Appendix C. Please refer to the supplemental video for visualization of the real world tasks: https://sites.google.com/view/robotics-datasets-analysis.

4 Probing Dataset Biases

Our investigation evaluates 5 unique datasets as candidates for pre-training (see Fig. 2, left) – ImageNet [2], Ego4D [15], 100 Days of Hands [17] (DoH), Kinetics [16], and RoboNet [32] (see Sec 4.1 for descriptions). These datasets were chosen because they are widely used in both computer vision and robotics, and because they have radically different compositions/purposes. Our experiments repeatedly apply our experimental methodology (see above) on various sub-samplings/combinations of this data. First, we evaluate dataset performance in isolation in order to understand which of them contain the most useful priors for robotics (see Sec. 4.1). However, we would like to achieve maximum performance, by training on as many diverse data sources as possible. So our next experiments analyze how well various combinations of the data perform (see Sec. 4.2). Finally, we investigate

	Task	Single Dataset Models (1M Images)					Baselines		
		ImageNet	Ego4D	Kinetics	100 DoH	RoboNet	Scratch	VC-1 [13]	MVP [18]
Sim	RoboSuite [36]	62	61	65	52	58	2	63	51
	MetaWorld [39]	90	93	87	86	74	72	70	83
	Franka Kitchen [40]	63	68	55	62	62	40	61	61
	Average (Sim)	72	74	69	67	65	38	65	65
Real	Block Stacking	68	64	72	55	76	0	60	52
	Pouring	44	19	50	19	13	0	25	13
	Toasting	10	0	10	40	0	0	10	10
	Average (Real)	41	28	44	38	30	0	33	25

Table 1: This table compares our pre-trained representations, which are each trained on 1M images from different datasets, against each other on our task suite. To provide additional context, we also evaluate against SOTA baselines from the field and a randomly initialized "Scratch" representation. Note that the ImageNet model provides surprisingly strong performance, both in simulation and in the real world!

dataset scaling, and find that the distribution of images matters a great deal during pre-training (see Sec. 4.3).

4.1 Evaluating Dataset Priors in a Vacuum

Before discussing this experiment in detail, we'd like to take a step back and analyze the subject datasets. (1) ImageNet is a classic computer vision dataset that contains 1000 train images for each of its 1000 classes. ImageNet's careful curation and incredible diversity may result in more expressive representations, but its focus on centered, high-quality images may be an ill fit for robotics. (2) Ego4D is a modern ego-centric video dataset, with 3.6K hrs of video collected by humans performing daily tasks in-the-wild. Ego4D's greatest strength is that it contains realistic images that a robot may observe in real-world environments. However, frames collected within the same video tend to look similar to each other, and the lack of curation mean that some object classes (e.g. blocks, cups, etc.) rarely appear. (3) DoH contains curated YouTube videos of humans manipulating various household objects (e.g. cooking video). The curation ensures that action classes are balanced and that videos look distinct from each other. Furthermore, the focus on manipulation may help the representations learn useful cues (e.g. where objects can be grasped). However, YouTube videos look very different from robot observations, so its unclear if any priors learned from DoH can transfer to robotics. (4) Kinetics(-700) is similar to DoH in that it's curated from YouTube, but its videos contain generic human interactions (e.g. with objects and/or other humans) instead of focusing on manipulation. (5) Finally, RoboNet contains 13M+ image observations of robots randomly interacting with objects placed in a bin in front of them. RoboNet could be invaluable for pre-training, since its images are highly relevant to the test-time domain. But robot data is collected in sterile lab environments, which could cause the representations to overfit to those exact settings.

It is clear that these five datasets have complex trade-offs that may affect their usability for robotics. The only way to settle the question is to compare them head-to-head. Thus, we apply our evaluation methodology to 1 Million frames sampled randomly from every dataset, in order to compare them apples-to-apples. This is easy to do in ImageNet, since it has 1M balanced train images. But for the video datasets (Ego4D/Kinetics/DoH), we first processed them into frames (sub-sampled at 3FPS) and then randomly select 1M images from the whole set. For RoboNet, we followed a similar procedure as used for the video datasets, but randomly sampled 1M image observations instead. This results in 5 representations that we evaluate on our task suite (via BC). For additional context, we also evaluate a "Scratch" model that has randomly initialized weights, alongside pre-trained weights downloaded from two SOTA baselines (VC-1 [13], MVP [18]). These baselines share our exact network architecture and pre-training method/hyperparameters, but were trained on many more frames (2.5M+), sampled primarily from Ego4D. The results are presented in Table 1.

Our first observation is that performance trends from simulation do not transfer to the real world at all (see Sec. 5 for more). Thus, we focus the rest of our analysis on the real world trends, since that's our main motivation as roboticists. The real robot experiments reveal that ImageNet/Kinetics/DoH representations all perform substantially better than those trained on RoboNet/Ego4D ($\sim 40\%$ v.s. 30% success rate). This is surprising, since the image content in the later datasets

	Task	Soup-1M	Soup 2M	Soup-1M + 1M Extra Frames (2M Total)				
				ImageNet	Ego4D	Kinetics	100 DoH	RoboNet
Sim	RoboSuite [36]	64	64	52	53	59	67	58
	MetaWorld [39]	87	89	92	86	87	92	88
	Franka Kitchen [40]	66	67	56	61	64	62	60
	Average (Sim)	72	73	67	67	70	74	69
Real	Block Stacking	76	44	72	60	76	92	76
	Pouring	38	38	32	38	32	38	32
	Toasting	10	40	0	10	22	50	0
	Average (Real)	41	41	35	36	43	60	36

Table 2: This table compares different data "soups" against each other. Soup-1M and Soup-2M are trained on 200K/400K images randomly sampled for the 5 target datasets. The models on the right are trained with the Soup 1M images and an additional 1M frames from the target dataset. For example, Soup 1M + Ego4D has the original 1M soup images, plus an additional 1M Ego4D frames. The soup experiments show that the image distribution matters more than the number of images trained on: e.g. Soup-2M does not improve on Soup-1M, but Soup-1M + 1M DoH does!

seems like a much closer match to the test tasks – e.g. RoboNet entirely contains images of robot interactions. But there is a pattern! Note that ImageNet/Kinetics/DoH were all sampled and curated from the internet (using YouTube/image search), and thus contain much greater frame diversity than the headset videos and robot trajectories in Ego4D/RoboNet respectively. We theorize that this increased diversity results in a stronger representation that is better equipped to generalize to entirely new environments during test time.

4.2 Combining Data from Different Sources

Another surprising result from Table 1 is that the baselines representations perform worse than the ImageNet/Kinetics/DoH representations, despite being trained on significantly more images. In fact, VC-1 actually pre-trains on ImageNet alongside many more ($2.5M+$) images from Ego4D, while using the exact same pre-training algorithm and hyperparameters that we do. A simple explanation for this discrepancy is that the VC-1 representation is functionally very similar to our Ego4D representation, since the majority of its pre-training frames comes from that dataset. Thus, every batch seen by the encoder during pre-training primarily consists of Ego4D frames. The key insight is the *distribution* of the pre-training set matters more than the raw number of frames trained on. This theory is supported by the experiments, since the VC-1 representation performs only marginally better than Ego4D (33% vs 28%).

So what is the right way to train across multiple datasets, if you don't know (a priori) which one is most useful for robotics tasks? A simple idea is to proportionally mix the datasets so that the model is pre-trained on an equal number of frames from each dataset. To test this idea, we create a "Soup-1M" pre-training set that contains $200K$ images randomly sampled from each of the 5 datasets. We then evaluate this model on our test suite (see Table 2, left). Note that the Soup-1M model performs about the same as the ImageNet/Kinetics/DoH models (41%), even though it was trained on a significant amount of (comparably weaker) Ego4D/RoboNet frames. This suggests that scaling to multiple datasets can increase robustness, so long as the datasets are kept carefully balanced (to maximize diversity) during pre-training.

4.3 Analyzing the Marginal Value of Each Dataset

While Soup-1M provides a sensible starting point for multi-dataset pre-training, it's not able to significantly exceed ImageNet/Kinetics/DoH in terms of task performance. This is reasonable, since all of these models were trained on 1M frames. However, it's unclear if simply increasing the size of the soup (e.g. training on 1M frames from each dataset) will substantially increase performance. After all, a key theme of the experiments so far is that the image distribution matters a great deal for successful pre-training. Thus, in this final experiment we seek to understand the marginal value of adding additional data to the soup.

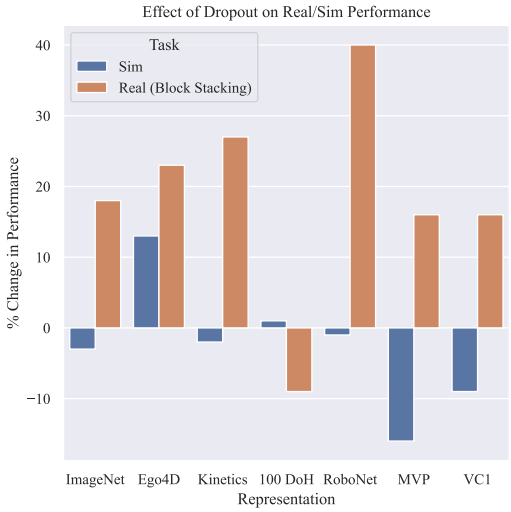

Figure 3: We compare policies trained using dropout on our simulation and real world (block stacking) tasks, against models trained without dropout. Note how dropout usually harms performance in simulation, whereas it provides a strongly positive effect in the real world evaluation!

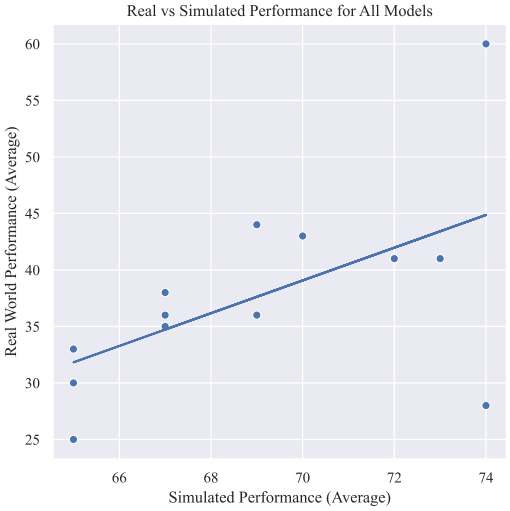

Figure 4: We plot average model performance in the real world against average performance in sim for all the models testing in our study. Note how the sim scores are only weakly predictive of real world performance ($R^2 = 32\%$). In fact, the best performing model gets the same score in sim as the lowest performer!

First, we train a Soup-2M model (identical to Soup-1M but w/ $400K$ images per/dataset) that tests scaling the Soup-1M model naively. Then we train 5 additional models on datasets containing Soup-1M and 1M additional images from a subject dataset (e.g. Soup-1M + 1M ImageNet frames, Soup-1M + 1M Ego4D frames, etc.). This represents both increasing the number of training frames, while shifting the train distribution towards that of the subject dataset. All six models are evaluated and results are present in Table 2 (right).

Our first take-away is that data scaling is indeed more nuanced than simply expanding the number of frames. After all, the Soup-2M model performs marginally better in simulation than SOUP-1M, and performs exactly the same (on average) in the real world. In contrast, the strongest model, Soup-1M + 1M DoH, is able to perform 20% better than Soup-1M (and 30% better than the strongest baseline) on the real world tasks! Finally, the Soup-1M + 1M Ego4D/ImageNet/RoboNet models perform slightly *worse* than Soup-1M, whereas the Soup-1M + 1M Kinetics model performs slightly better. These results are mostly in line with our expectations from Sec. 4.1 (e.g. adding more RoboNet data reduces performance, while adding more Kinetics/DoH increases performance). Thus, we can conclude that some of our basic intuitions (e.g. importance of diversity over content) hold, even when adding frames to a pre-existing image pool.

5 Ablating our Experimental Setup

While testing and implementing our experimental methodology (see Sec. 3), we found some insights, which may be useful for the community. First, we found that simple regularization techniques, like adding Dropout [19] to the BC policy network, significantly improved our real world policies' robustness. Second, we noticed that some important trends do not transfer from the sim tasks to real world. The following paragraphs further analyze these observations.

Regularizing Policies w/ Dropout Early in our evaluations, we noticed that the policies often produced jerky motions that could damage the robot and its environment. Thus, we searched for a simple fix that could improve robustness in the real world. We found that adding dropout to the policy network (w/ $p = 0.2$) significantly improved the robot's qualitative behavior: the commanded motions became smoother, and it was able to generalize better to new scenarios. This improvement was quantified on the *Block Stacking* task, by fine-tuning our five 1M models (see Sec. 4.1) and the baselines w/ and w/out dropout. Note how adding dropout almost uniformly improves policy

performance (see Fig. 3, orange bars) in the real world, but it has the opposite effect in sim (more on that later)!

Our intuition was that dropout helped the policy become robust to the stochastic dynamics/disturbances it will encounter in the real world. To test this hypothesis, we create a new task, named *Block Stacking Robust*, to evaluate policy robustness. In *Block Stacking Robust*, the human pushes the cube out of the robot's gripper in the middle of the episode (i.e. right before the robot grasps). This forces the robot to dynamically replan its actions, and respond to a scenario it never saw during training. We find that the average success rate on this task (across all models) increases to **24**$\%$ from 10% thanks to dropout! This substantial improvement (on a challenging task) shows the importance of regularization for increased robustness during test-time.

Analyzing Sim-to-Real Transfer You may have noticed a consistent theme throughout our experiments so far: trends in simulation are almost entirely disconnected from real world performance. For example, the simulation suite predicts that Ego4D is the best representation for robotics, but the real world results consistently disagree with that assessment (see Table 1). Furthermore, key implementation details in the real world (like Dropout) can actually hurt performance in simulation (see how Dropout hurts sim performance in Fig. 3). It's not just that simulation missed a few key trends: we plot the sim performance vs real performance for all models in Fig. 4 and find that the two values are almost entirely uncorrelated ($R^2 = 32\%$). This result is not necessarily surprising, since the sim2real gap in manipulation is well known. However, we feel it is important to explicitly do this analysis, since it's still very common practice in prior work [1, 13, 18, 14, 42] to draw inferences about pre-trained representations using simulated benchmarks (e.g. CortexBench [13], Franka Kitchen [40], Isaac Gym [18], etc.).

6 Discussion

This paper investigated how dataset biases and implementation choices can affect visual pre-training for robotics. Our experiments analyzed pre-trained MAE representations trained on 15 different data distributions, sourced from 5 common datasets. These models were evaluated on standard sim environments, alongside 3 diverse and challenging real world tasks ($50+$ robot evaluations). We find that standard computer vision datasets (e.g. ImageNet) provide surprisingly strong performance for robotic pre-training. In fact, our simple ImageNet representations outperforms both Ego4D representations and representative baselines in the field. They key insight is that the diversity of the image distribution, matters much more than the raw number of images during pre-training. Guided by this insight, we explore various strategies for scaling data distribution, by carefully mixing data from different sources. As part of this investigation, we train a final model (Soup-1M + 1M DoH) that exhibits a 30% improvement over the baselines on real world robotic tasks. Finally, we analyze our experimental methodology, and show how simple regularization techniques (e.g. Dropout) can boost real world performance, and conclude that simulation results do not transfer to real world tasks. We hope that our results will inspire others in the field to study how various sources of offline data can transfer to robotics tasks. To enable this effort, we will release all of our pre-trained representations and evaluation code on our website[2].

Limitations and Future Work While our experiments were extensive, there are some limitations that should be addressed by future work. First, there is no simple theory or experimental test that can predict if a representation will actually work well on the robot after pre-training. In fact, our experiments show that the easiest evaluation technique, sim, gives a sense of false progress in the real world! Thus, it is vital for the field to find faster ways to evaluate representations, and share results. One possibility is a standardized cloud robotics benchmark [43, 44] that could greatly reduce the load for researchers. Next, our experiments focused on a specific methodology (MAE pre-training + BC fine-tuning) that was closely tied to prior work in the field. However, it's unclear if the trends we find in this setting, will transfer to other tasks/settings (e.g. RL). Thus, it would be interesting to replicate our study in other scenarios to see if our findings still hold. Finally, it would be interesting to expand our analysis to other robotic tasks, like visual navigation and grasping.

[2] https://sites.google.com/view/robotics-datasets-analysis

References

[1] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[4] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[5] 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.

[6] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *AISTATS*, 2010.

[7] S. Haldar, J. Pari, A. Rai, and L. Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023.

[8] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[9] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[10] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[13] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? 2023.

[14] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[15] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[16] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman. A short note on the kinetics-700-2020 human action dataset. *arXiv preprint arXiv:2010.10864*, 2020.

[17] D. Shan, J. Geng, M. Shu, and D. Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[18] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

[20] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3 (6):233–242, 1999.

[21] 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*, 31, 2018.

[22] 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.

[23] 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*, 37(4-5):421–436, 2018.

[24] 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)*, pages 3406–3413. IEEE, 2016.

[25] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[26] 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.

[27] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[28] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

[29] S. Dasari and A. Gupta. Transformers for one-shot visual imitation. In *Conference on Robot Learning*, pages 2071–2084. PMLR, 2021.

[30] W. Whitney, R. Agarwal, K. Cho, and A. Gupta. Dynamics-aware embeddings. *arXiv preprint arXiv:1908.09357*, 2019.

[31] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[32] 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. *arXiv preprint arXiv:1910.11215*, 2019.

[33] 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*, pages 12104–12113, 2022.

[34] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[35] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Handbook of robotics*, 59(BOOK_CHAP), 2008.

[36] 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 *Conference on Robot Learning (CoRL)*, 2021.

[37] R. Rahmatizadeh, P. Abolghasemi, A. Behal, and L. Bölöni. From virtual demonstration to real-world manipulation using lstm and mdn. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[38] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[40] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning (CoRL)*, 2019.

[41] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[42] N. Hansen, Z. Yuan, Y. Ze, T. Mu, A. Rajeswaran, H. Su, H. Xu, and X. Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. In *International Conference on Machine Learning (ICML)*, 2023.

[43] S. Dasari, J. Wang, J. Hong, S. Bahl, Y. Lin, A. S. Wang, A. Thankaraj, K. S. Chahal, B. Calli, S. Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[44] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, et al. Train offline, test online: A real robot learning benchmark. *arXiv preprint arXiv:2306.00942*, 2023.