For Pre-Trained Vision Models in Motor Control, Not All Policy Learning Methods are Created Equal

Yingdong Hu [1 2 3] **Renhao Wang** [1] **Li Erran Li** [4 †] **Yang Gao** [1 2 3]

Abstract

In recent years, increasing attention has been directed to leveraging pre-trained vision models for motor control. While existing works mainly emphasize the importance of this pre-training phase, the arguably equally important role played by downstream policy learning during control-specific fine-tuning is often neglected. It thus remains unclear if pre-trained vision models are consistent in their effectiveness under different control policies. To bridge this gap in understanding, we conduct a comprehensive study on 14 pre-trained vision models using 3 distinct classes of policy learning methods, including reinforcement learning (RL), imitation learning through behavior cloning (BC), and imitation learning with a visual reward function (VRF). Our study yields a series of intriguing results, including the discovery that the effectiveness of pre-training is highly dependent on the choice of the downstream policy learning algorithm. We show that conventionally accepted evaluation based on RL methods is highly variable and therefore unreliable, and further advocate for using more robust methods like VRF and BC. To facilitate more universal evaluations of pre-trained models and their policy learning methods in the future, we also release a benchmark of 21 tasks across 3 different environments alongside our work. Source code and more details can be found at https://yingdong-hu.github.io/PVM-control/.

[1]Tsinghua University [2]Shanghai Artificial Intelligence Laboratory [3]Shanghai Qi Zhi Institute [4]AWS AI, Amazon. [†] Work done outside of Amazon. Correspondence to: Yang Gao <gaoyangiiis@tsinghua.edu.cn>, Yingdong Hu <huyd21@mails.tsinghua.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

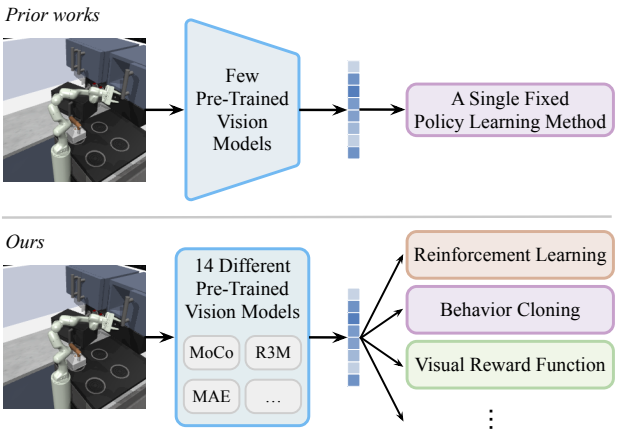


Figure 1: **(Top)** Most prior works that leverage pre-trained vision models as frozen perception modules for motor control only compare a few models using a single fixed policy learning algorithm. **(Bottom)** We find that using different policy learning algorithms results in significant changes in the rankings of 14 different vision models, i.e., the effectiveness of a vision model is algorithm-dependent.

1. Introduction

The transfer of pre-trained features to various parallel and even orthogonal downstream tasks is a ubiquitous paradigm in modern deep learning. This well-explored approach has revolutionized computer vision (Chen et al., 2020a; He et al., 2020), natural language processing (Devlin et al., 2018; Brown et al., 2020), and other fields (Baevski et al., 2020). The transferability of visual features has been thoroughly demonstrated (Tan et al., 2018), leading to an explosion of pre-trained models in computer vision.

A growing body of research is focused on rendering such pre-trained models as the cornerstones for vision-based motor control (Parisi et al., 2022; Xiao et al., 2022; Radosavovic et al., 2022; Nair et al., 2022; Ma et al., 2022). Since these models have typically been trained on large-scale natural visual data, their features possess general knowledge about the semantics of our world and its properties, which is invaluable for universal control. A downstream set of modules then learn a control policy by adapting these out-of-domain features to in-domain control data.

However, the emphasis to date has largely been on how the pre-training phase can be done better. To learn a downstream control policy, prior works often make ad-hoc choices around reinforcement learning or imitation learning approaches (Xiao et al., 2022; Parisi et al., 2022), resulting in a lack of understanding of the impact of downstream control policy on final performance. Can a given pre-trained vision model maintain consistent effectiveness across different downstream policy learning methods? If not, can we explain the difference? Given multiple possible downstream policy learning methods, how should we evaluate a pre-trained vision model?

To answer these questions, we conduct a large-scale benchmarking study on diverse pre-trained vision models for a plethora of control tasks across different environments. Our study treats the often-neglected policy learning phase as a first-class citizen. As shown in Figure 1, we consider *three* policy learning algorithms: (*i*) reinforcement learning (RL), (*ii*) imitation learning through behavior cloning (BC), and (*iii*) imitation learning with a visual reward function (VRF). The first two approaches (RL and BC) are widely used in the existing literature and treat pre-trained features as representations that encode environment-related information. The last approach (VRF) is an inverse reinforcement learning (IRL) paradigm we adopt which requires that the pre-trained features also capture a high-level notion of task progress, an idea that remains largely underexplored. We then consider 14 pre-trained vision models covering different architecture (ResNet; He et al. 2016 and ViT; Dosovitskiy et al. 2020) and prevalent pre-training methods (contrastive learning; Chen et al. 2021b, self-distillation; Caron et al. 2021, language-supervised; Radford et al. 2021, masked image modeling; Bao et al. 2021, etc.). For a fair and comprehensive comparison, we run extensive experiments on 21 simulated tasks across 3 robot manipulation environments: Meta-World (Yu et al., 2020), Franka-Kitchen (Gupta et al., 2019), and Robosuite (Zhu et al., 2020). Our investigation reveals surprising results and contributions:

- **Lack of consistently performant models.** The effectiveness of a pre-trained vision model is highly dependent on the downstream policy learning method.
- **Point out directions for reliable evaluation methods.** Due to high variability, RL is not a robust evaluation method. We show that the consistent results of VRF and BC make them reliable evaluation methods in our benchmark of pre-trained vision models for motor control.
- **Deeper dive into properties of vision models** enables us to obtain metrics, such as linear probing loss and k-NN classification accuracy, that have substantive predictive power for downstream control policies.

2. Related Work

Pre-training in computer vision. Large-scale pre-training has become the new fuel empowering computer vision. Contrastive learning and related methods (Hadsell et al., 2006; Wu et al., 2018; Caron et al., 2020; Hu et al., 2022) learn visual representations by modeling image similarity (Grill et al., 2020) and dissimilarity (Chen et al., 2020a) between two or more views. Masked Image Modeling (MIM) (Bao et al., 2021) pursues a different direction by learning to predict removed pixels (He et al., 2022), discrete visual tokens (Peng et al., 2022), or pre-computed features (Wei et al., 2022). Language-supervised pre-training, e.g., CLIP (Radford et al., 2021) and related works (Mu et al., 2022; Dong et al., 2022), has been established as a powerful paradigm for learning visual representations. While pre-trained models attract increasing attention in the vision field, no large-scale evaluation has compared the various models available for motor control. This work aims to benchmark the plethora of pre-trained vision models to explore which ones are the most effective for visuomotor control.

Pre-trained vision models for motor control. The application of pre-trained vision models to problems in motor control is a rapidly growing field (Radosavovic et al., 2022; Wang et al., 2022), with studies such as RRL (Shah & Kumar, 2021), PIE-G (Yuan et al., 2022), and MVP (Xiao et al., 2022) demonstrating the effectiveness of supervised or self-supervised pre-trained vision models as visual representations for RL agents. PVR (Parisi et al., 2022) and R3M (Nair et al., 2022) find that vision models pre-trained on real-world data enable data-efficient behavior cloning on diverse control tasks. VIP (Ma et al., 2022) proposes a self-supervised pre-trained vision model capable of producing dense reward signals. Concurrently, Hansen et al. (2022) show that a carefully designed Learning-from-Scratch (LfS) baseline is competitive with methods that leverage pre-trained vision models. However, most approaches train the agent with only BC or only RL, with limited or no discussion on how policy learning choices are made. Thus, it remains unclear whether the effectiveness of pre-trained vision models is consistent across different policy learning methods.

Policy learning. Reinforcement learning (RL) (Sutton & Barto, 2018) and imitation learning (IL) (Hussein et al., 2017) are two mainstream approaches for policy learning. The gap between image-based RL and state-based RL has been significantly bridged, largely due to ideas like autoencoder-based architectures (Hafner et al., 2019; Yarats et al., 2021b), self-supervised objectives (Laskin et al., 2020b; Schwarzer et al., 2020), and data augmentation (Kostrikov et al., 2020; Laskin et al., 2020a). IL can be broadly categorized into Behavior Cloning (BC) (Pomerleau, 1988) and Inverse Reinforcement Learning (IRL) (Ng et al., 2000). BC is extremely sample-efficient but may suf-

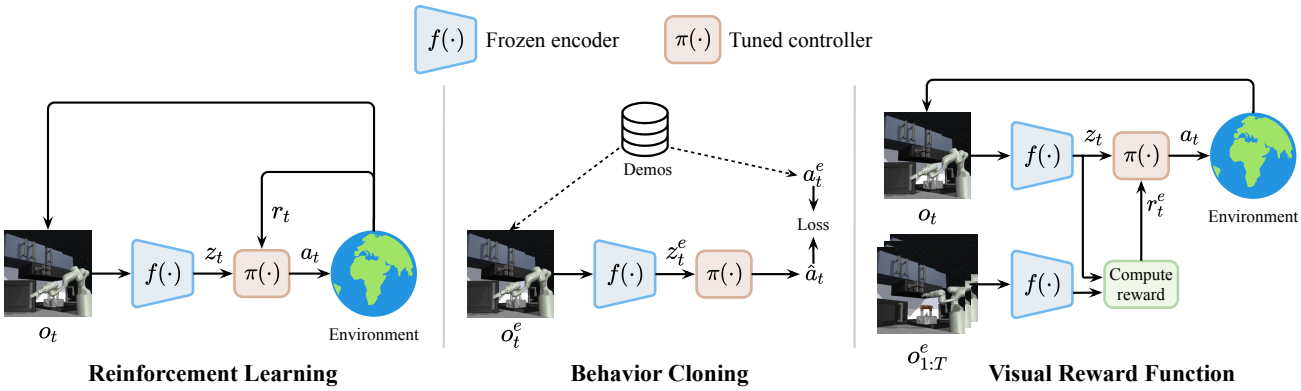


Figure 2: Illustration of three downstream policy learning methods we considered. From left to right: reinforcement learning (RL), imitation learning through behavior cloning (BC), and imitation learning with a visual reward function (VRF).

fer on out-of-distributions samples (Ross et al., 2011) or copycat problems (Wen et al., 2020). IRL focuses on learning a robust reward function (Kostrikov et al., 2018). In this work, we aim to contrast the merits of three different policy learning methods (i.e., RL, BC, IRL) and their properties with respect to re-appropriating pre-trained general vision models for downstream control-specific problems.

3. Approach

In this section, we cover the different components of our study, beginning by describing the policy learning methods we considered in Sec. 3.1, 14 pre-trained vision models in Sec. 3.2, and 3 simulation environments in Sec. 3.3.

3.1. Policy Learning Methods

In general, we consider agents acting within a standard Markov Decision Process (MDP), where at each time step we have access to a tuple $(\mathcal{O}, \mathcal{A}, P, R, \gamma)$, replete with the usual definitions. The agent for learning motor control consists of an encoder network f and a lightweight controller head π. The encoder f is a *frozen* pre-trained vision model. Given an image observation o_t from the environment, f first extracts a representation vector $z_t = f(o_t)$. Then, the controller π takes in the representation and predicts an action $a_t = \pi(z_t)$. We analyze the following three representative policy learning methods and illustrate them in Figure 2.

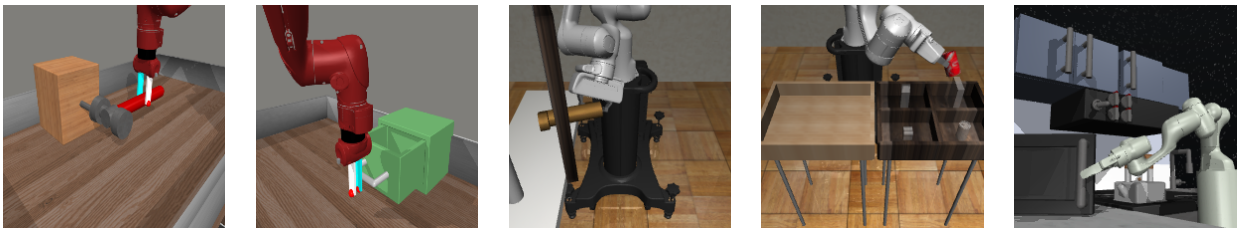
Reinforcement learning. In model-free RL, the goal is to use the in-domain experience to maximize the expected discounted sum of rewards $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R\left(o_t, a_t\right)\right]$. Specifically, we use DrQ-v2 (Yarats et al., 2021a), a state-of-the-art off-policy actor-critic approach for continuous vision-based control. DrQ-v2 is a representative and also useful choice, given its favorable sample efficiency. Additional algorithmic details are available in Appendix A.1.

Imitation learning through behavior cloning. Given access to expert trajectories $\mathcal{T}^e = \{(o_t^e, a_t^e)_{t=0}^T\}_{n=0}^N$, BC corresponds to minimizing the loss function $\mathbb{E}_{(o_t^e, a_t^e)\sim\mathcal{T}^e} \left\|a_t^e - \pi(f(o_t^e))\right\|_2^2$. Both RL and BC are common approaches to evaluate vision models as frozen perception modules for policy learning in literature. The main role played by the vision model here is to extract representations that contain environment-relevant information. These visual representations are used to replace hand-engineered ground-truth features, which are hard to estimate across diverse control tasks.

Imitation learning with a visual reward function. Another method to tackle the imitation learning problem involves Inverse Reinforcement Learning (IRL) (Ng et al., 2000). IRL infers the underlying reward function from the expert trajectories before employing RL to optimize a policy. In our setting, the crucial idea is to craft the underlying reward function based on a distance metric in the vision model's embedding space. We term this the *visual reward function (VRF)*. For example, one straightforward strategy is to define the reward as the negative squared ℓ_2 distance between the agent's observation and the expert goal image: $-\left\|f(o_t) - f(o_T^e)\right\|_2^2$ (Zakka et al., 2022). In our experiments, we adopt a recent algorithm ROT (Haldar et al., 2022), which derives the reward based on the Sinkhorn distance (Cuturi, 2013) between the expert and the agent observations (detailed in Appendix A.2). VRF requires not only that the vision model faithfully substitutes ground-truth features, but also that it encodes task progress information in its latent representations. The effectiveness of pre-trained vision models in VRF is underexplored. But such an approach holds great potential since it enables agents to learn directly from diverse human videos (Chen et al., 2021a; Kumar et al., 2022).

3.2. Pre-Trained Vision Models

We aim to investigate the efficacy of different "off-the-shelf" pre-trained vision models for motor control. We consider



Hammer | Drawer Close | Panda Door | Panda Pick-and-Place | Opening Microwave

Figure 3: Example tasks from 3 environments. Due to the space limit, we only show 2 tasks from Meta-World, 2 tasks from Robosuite, and 1 task from Franka-Kitchen. Please see Appendix C for all 21 tasks.

Table 1: The highlights of different pre-trained models.

Model	Highlights
MoCo v2	Contrastive learning, momentum encoder
SwAV	Contrast online cluster assignments
SimSiam	Without negative pairs
DenseCL	Dense contrastive learning, learn local features
PixPro	Pixel-level pretext task, learn local features
VICRegL	Learn global and local features
VFS	Encode temporal dynamics
R3M	Learn visual representations for robotics
VIP	Learn representations and reward for robotics
MoCo v3	Contrastive learning for ViT
DINO	Self-distillation with no labels
MAE	Masked image modeling (MIM)
iBOT	Combine self-distillation with MIM
CLIP	Language-supervised pre-training

the following 14 models across 2 architectures.
ResNet-50: MoCo v2 (Chen et al., 2020b), SwAV (Caron et al., 2020), SimSiam (Chen & He, 2021), DenseCL (Wang et al., 2021), PixPro (Xie et al., 2021), VICRegL (Bardes et al., 2022), VFS (Xu & Wang, 2021), R3M (Nair et al., 2022) and VIP (Ma et al., 2022).
ViT-B/16: MoCo v3 (Chen et al., 2021b), DINO (Caron et al., 2021), MAE (He et al., 2022), iBOT (Zhou et al., 2021) and CLIP (Radford et al., 2021).

Table 1 summarizes the highlights of different models, and Appendix B holds detailed descriptions. We choose these models for their diversity and coverage of the pre-trained model landscape: *(i)* R3M and VIP are designed with robotic manipulation tasks in mind. *(ii)* DenseCL, PixPro and VICRegL learn local visual features and we speculate that they may benefit motor control, which requires fine-grained spatial information. *(iii)* Other models serve as excellent references given their documentation in previous works, e.g., MoCo v2 and MAE are used in PVR (Parisi et al., 2022) and MVP (Xiao et al., 2022), respectively.

All models have official open-source codebases, from where we obtain pre-trained weights. One variable that is difficult to control is the pre-training dataset, as it is prohibitively expensive to retrain all the models we considered on the same dataset. However, the pre-training dataset is not a core factor affecting downstream control tasks, as evidenced by Parisi et al. (2022). In addition, all the models are pre-trained on out-of-domain data, i.e., they have never seen a single in-domain image from the environment. Thus, all our subsequent comparisons are on equal footing.

3.3. Environments

Selection criteria. Our core criteria for selecting benchmark environments is that they are representative of real-world scenarios. An excellent environment should *(i)* support low-level full-physics control (i.e., no magic skills/abstract action space), *(ii)* render visually-realistic observations, and *(iii)* cover diverse tasks and objects. Additionally, dense rewards need to be provided to study RL algorithms, and fast simulation speeds up experimentation. Some environments used in previous works fall short on one or more of our requirements. For example, Habitat (Savva et al., 2019) does not support full-physics simulation. DeepMind Control (DMC) Suite (Tassa et al., 2018) focuses on locomotion tasks, and its observations are not visually realistic.

Three environments. Taking the above factors into consideration, we use a total of 21 tasks across 3 robot manipulation environments: Meta-World (8 tasks), Robosuite (8 tasks), and Franka-Kitchen (5 tasks). Figure 3 shows sample tasks from each environment. Complete environment details are available in Appendix C. All environments are simulated via the MuJoCo physics engine (Todorov et al., 2012), which enables fast simulation of physical contact. We choose distinct challenge tasks from the environments covering various scenes, objects and manipulation skills.

3.4. Experimental Setup

Implementation details. All environment observations are 224×224 RGB images, which are consistent with the resolution most vision models are pre-trained at. In our experiments, we find that using only one image observation is comparable with using a stack of consecutive images, so we choose the more compute-efficient option. Furthermore, we do not use proprioceptive information (e.g., end-effector poses and joint positions, etc.), ensuring fair comparison of the vision models operating strictly with visual observations. For more details, we refer to Appendix D.1.

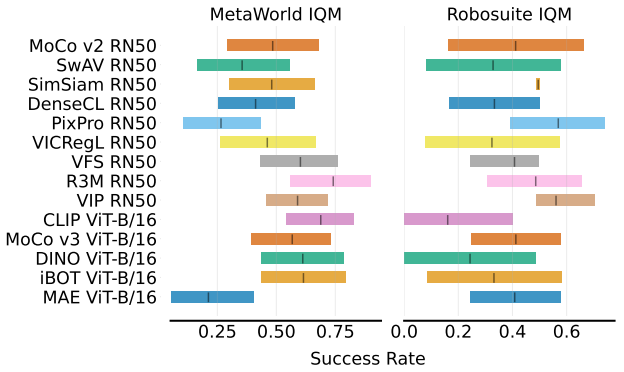


Figure 4: Aggregate RL performance on Meta-World and Robosuite with 95% CIs based on 8 tasks per environment. There is no consistency between the two environments.

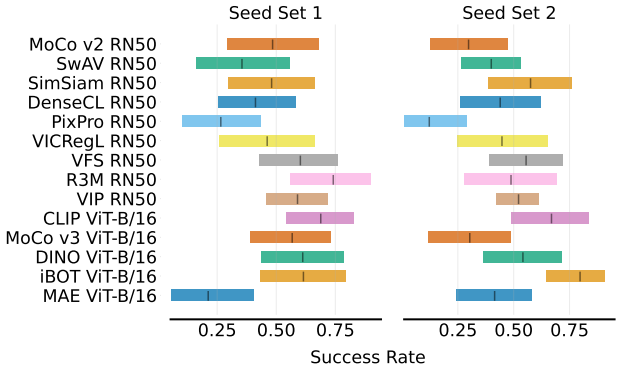


Figure 5: Aggregate RL performance (IQM scores) of two seed sets on Meta-World. The only difference between the two experiments is the seeds.

Evaluation. For each pre-trained vision model and each task, we run 3 seeds of BC and VRF, and 6 seeds of RL (due to its higher variability). After each run, we compute the policy success rate over 100 online rollouts. For the final aggregate performance across tasks, we report interquartile mean (IQM), which is not affected by outliers and has smaller uncertainty (the median and mean scores are shown in Appendix E.2). Following the guidelines of Agarwal et al. (2021), we also report interval estimates via stratified bootstrap confidence intervals (CIs) to further account for uncertainty in results.

4. Experimental Results

In this section, we first separately analyze the performance of pre-trained vision models for the three policy learning methods. Then, we further investigate what properties of vision models matter for different policy learning methods.

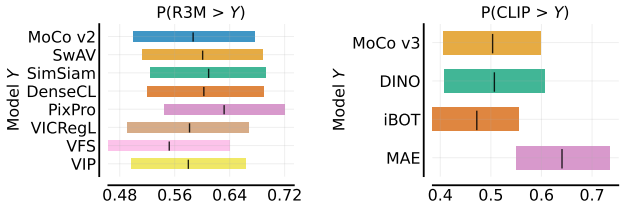


Figure 6: Probability of improvement on Meta-World with 95% CIs. Each subplot shows the probability of improvement of a given vision model compared to all other models.

4.1. Reinforcement Learning

Inconsistency between environments. Figure 4 shows the RL results of all pre-trained vision models on Meta-World and Robosuite. Surprisingly, the rankings of different vision models on the two environments are entirely different. In particular, PixPro ranks second to last on Meta-World but is one of the best-performing models on Robosuite. The language-supervised CLIP, which performs extremely well on Meta-World, is the worst one on Robosuite. At first glance, the effectiveness of a pre-trained vision model seems environment-dependent.

Inconsistency between runs. We further observe that the inconsistency exists not only in different environments, but even in different training runs in the same environment. Specifically, we repeat the experiment on Meta-World using a different set of 6 seeds. The results are shown in Figure 5. For most models, there is a huge difference in the success rate between the two experiments, with the largest being around 25% (R3M). This leads to the fact that even considering only one environment, we cannot reliably conclude which vision model is the best.

High variability. We hypothesize that the inconsistency between environments and runs is due to the substantial variability of current RL algorithms. This can be confirmed by the significantly large confidence intervals (CIs) in Fig. 4 and Fig. 5. At the same time, the CIs strikingly overlap for most pre-trained vision models, making it difficult to compare any two models directly. Hence, we show the *probability of improvement* metric proposed by Agarwal et al. (2021) in Fig. 6. On Meta-World, there is only a 40 - 60% chance that CLIP improves upon MoCo v3, although CLIP outperforms MoCo v3 by about 20% in point estimates of IQM (Fig. 4, left). We note that the high variability of RL results is also observed in recent works (Henderson et al., 2018; Chan et al., 2019; Xiao et al., 2022). This high variability is due to inherent randomness, which can arise from exploratory choices made during training, stochasticity in the task, and randomly initialized parameters.

Not an ideal evaluation method. In addition to exhibiting substantial variability, RL is notorious for being sensitive to lower-level choices (Andrychowicz et al., 2020) like hyper-

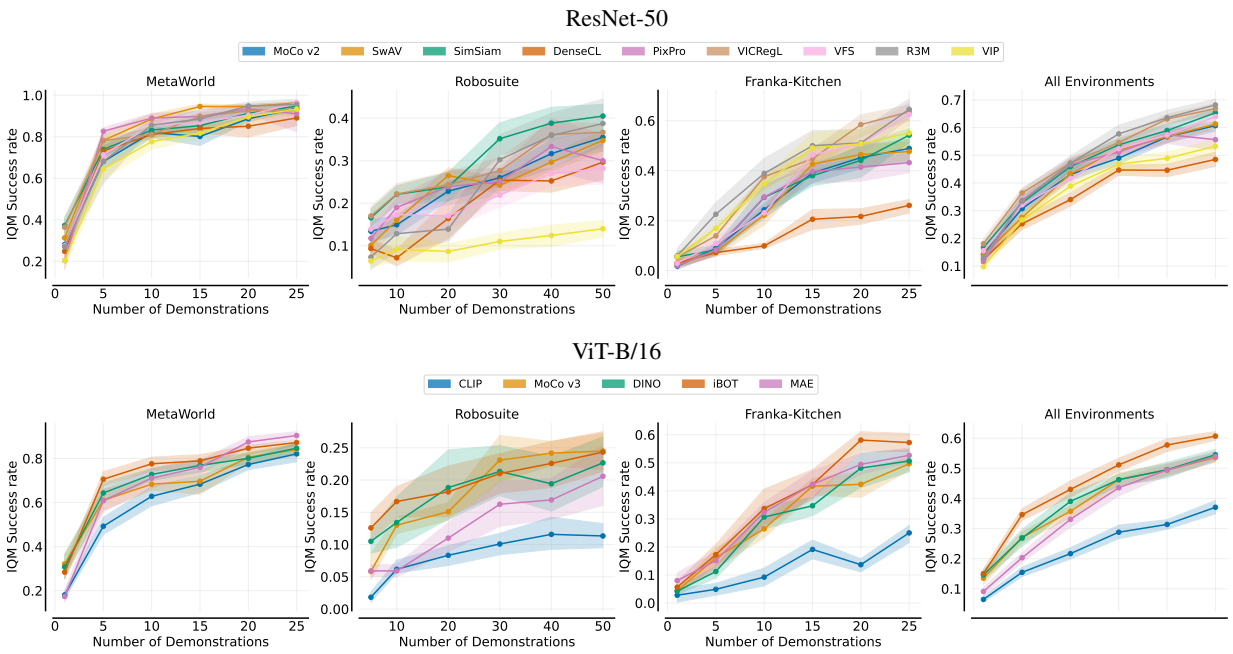


Figure 7: BC performance of different pre-trained vision models as a function of the number of demonstrations measured via IQM success rate. Shaded regions show pointwise 95% percentile stratified bootstrap CIs.

parameter selection. We use the exact same hyperparameters for all vision models, but we find that minor changes to some hyperparameters (e.g., learning rate, buffer size) cause significant differences for a given model. All these findings suggest that RL itself is *not* suitable as a downstream policy learning method to evaluate different pre-trained vision models. Further analysis on uncertainty is in Appendix E.1.

4.2. Imitation Learning through Behavior Cloning

We now thoroughly evaluate the large suite of pre-trained vision models using BC. We train BC policies with varying numbers of expert demonstrations to study how dataset size impacts performance in the low data regime. The results in Figure 7 demonstrate that no individual pre-trained vision model can dominate all three environments. However, considering 21 tasks across all environments (Fig. 7 rightmost), R3M is the best-performing model. This echoes the finding of Nair et al. (2022) that R3M enables data-efficient behavior cloning for robotic manipulation. Furthermore, there is a model that follows R3M closely: VICRegL. Interestingly, VICRegL is pre-trained purely on ImageNet (Russakovsky et al., 2015), without robotic manipulation tasks in mind, while R3M is pre-trained on diverse human videos. This lends credence to a dominant paradigm that advancements in vision can potentially be transferred directly to visuomotor control without extensive control-specific adaptation.

With respect to architectures, iBOT is clearly the best-performing model among ViTs, highlighting the benefit of combining masked image modeling with self-distillation. Another general observation is that ResNet-based models tend to outperform ViT-based models. This is in spite of the fact that model size and computation complexity tend to favor ViTs. We hypothesize this is because transformer-based architectures contain less visual prior knowledge.

In addition to robust empirical performance, BC is also very simple and sample-efficient. BC shows a high correlation with linear probing results (see Sec. 4.4), which can reflect whether the visual features encode environment-related information. These all verify the reliability of using BC to evaluate different pre-trained vision models.

4.3. Imitation Learning with Visual Reward Functions.

Figure 8 shows the VRF results of different pre-trained vision models on three environments. There are several intriguing observations: *(i)* All performant models across the board (e.g., MoCo v2, VICRegL, DINO, etc.) share one commonality: they are based on *joint embedding architectures* (LeCun, 2022) that force the global features to be invariant to a sampling process selecting pairs of different views of the same image. *(ii)* Perhaps most surprisingly, MAE exhibits near *zero* performance on all three environments, which means that the rewards computed from MAE representations are entirely meaningless (detailed analysis in the next section). *(iii)* R3M, the star model on BC, only obtains 25% IQM success rate when considering all environments tasks, ranking as the second-worst model. *(iv)* Some

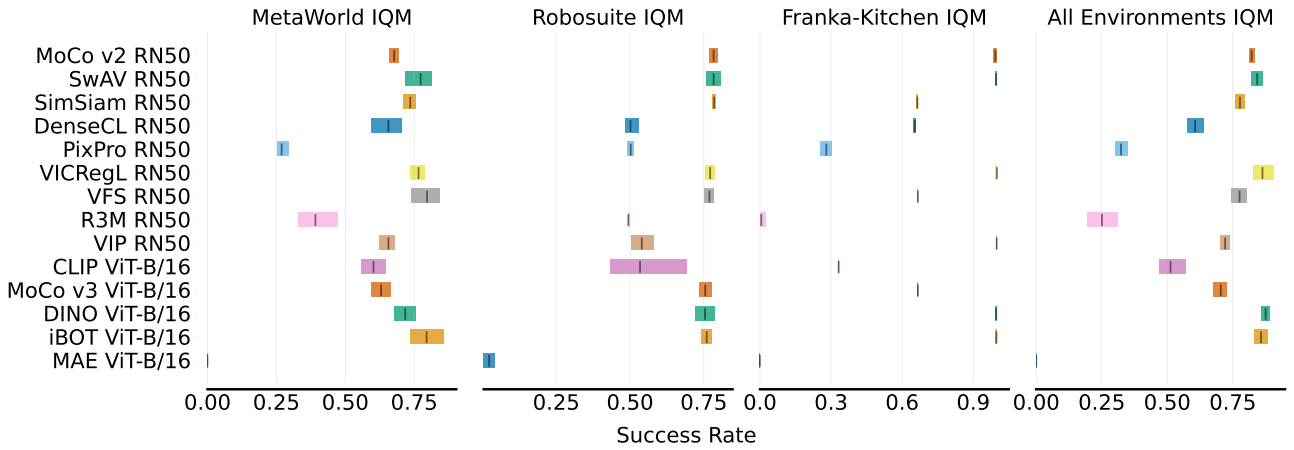


Figure 8: Aggregate VRF performance (IQM scores) of different pre-trained vision models with 95% CIs. The models rely on joint embedding architectures perform well. The results of MAE are close to *zero*.

models that emphasize learning local image characteristics (e.g., PixPro and DenseCL) also do not perform well.

Summarising these results, we see that the ideal pre-trained vision model that VRF requires is starkly different from the one that RL and BC need. Specifically, an excellent vision model for VRF should learn features at a global scale. Additionally, the invariance properties brought by joint embedding architectures outweigh the amount and type of pre-training data.

Notably, different pre-trained vision models yield very consistent performance when using VRF. Moreover, VRF can reveal whether the visual representations capture a notion of task progress. Finally, the policies trained with VRF use only one expert demonstration and achieve high performance. All these merits make VRF a strong candidate for evaluating different vision models for motor control.

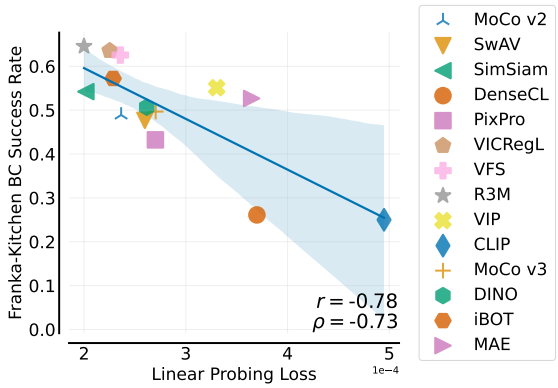


Figure 9: Correlation between Franka-Kitchen linear probing loss on the validation set and BC success rate (IQM scores) based on 5 tasks. Correlation coefficients (Pearson's r and Spearman's ρ) are shown in the bottom right.

4.4. Understanding Properties of Vision Models

The prior experiments focus on performance-driven comparisons of pre-trained vision models using different policy learning methods. But what specific properties of various vision models enable their comparative advantage?

Properties crucial for BC. The first question we set out to answer is what information is necessary for data-efficient behavior cloning. We borrow the commonly employed linear probing protocol in self-supervised visual representation learning (Zhang et al., 2016; Oord et al., 2018) to facilitate our analysis. Specifically, we train a single fully-connected layer on top of the frozen pre-trained features to predict the hand-engineered environment states. We train this linear regressor using images from the deployment environment by minimizing the mean squared error between the predictions and ground-truth state features. We use the validation loss as a proxy for the quality of visual features. Lower loss indicates that more environment-relevant information, such as object locations and joint positions, is retained in the pre-trained visual representations. Details are in Appendix D.2.

For Franka-Kitchen, we show the correlation coefficients between linear probing loss and BC success rate in Figure 9. We observe a strong *inverse correlation*, with Pearson's r being -0.78 and Spearman's ρ being -0.73. This indicates that pre-trained vision models that effectively encode ground-truth environment information will lead to more capable BC agents. Our findings also suggest that the linear probing protocol can be a valuable and intuitive alternative for evaluating vision models for motor control.

Properties crucial for VRF. We next explore what causes considerable distinctions in the VRF performance of different models. We first examine the relationship between hand-designed environment rewards and imitation rewards computed in the vision model's embedding space. The environment rewards capture human intuition for how a task

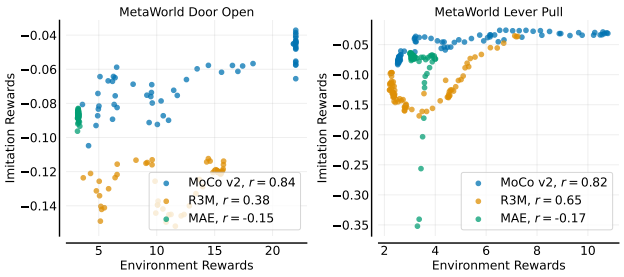
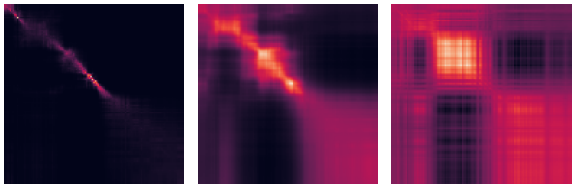


Figure 10: Environment rewards (x-axes) vs. imitation rewards correlation (y-axes). We plot the rewards collected using one online rollout after training.



MoCo v2 R3M MAE

Figure 11: OT matrix between two expert trajectories on Meta-World `door-open`. Brighter colors indicate higher values. This OT matrix is obtained by solving Equation 5.

should be solved. The scatterplots of three typical models on two Meta-World tasks are shown in Figure 10. The results show that the imitation rewards of MoCo v2 exhibit a much stronger correlation with the environment rewards, while there is little to no correlation for MAE. R3M falls somewhere in between. The uninformative rewards cause MAE to have near *zero* VRF scores on all control tasks.

To gain a deeper understanding of the reward correlation differences among vision models, we visualize the optimal transport (OT) matrix in the ROT algorithm. The OT matrix illustrates the correspondences/optimal alignment between two distributions. Specifically, we compute the OT between two expert trajectories. Since the two expert trajectories exhibit similar behaviors, the OT matrix computed from an ideal vision model should have high values on the approximate *diagonal*. The results are shown in Figure 11. We observe that the high values of MoCo v2 are concentrated on the diagonal, while the values of MAE are not, indicating that for a given image observation, nearly all the images in the other trajectory are equally similar. This suggests that MAE features cannot capture a notion of task progress.

We further find a metric that is highly predictive of VRF performance: ImageNet k-NN classification accuracy, as shown in Figure 12. The correlation between k-NN accuracy and VRF performance aggregated on all environments is as high as $r = 0.91$. Note that the poor performance of MAE on both VRF (near *zero*) and k-NN classification (27.4%) highlights the difficulty of using MAE features

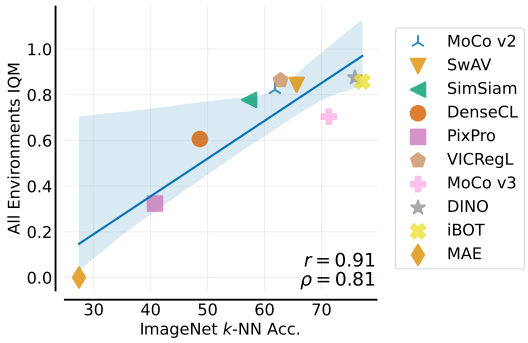


Figure 12: Correlation between ImageNet k-NN classification accuracy and VRF performance on all environments. We show all the vision models pre-trained on ImageNet.

directly through simple similarity metrics such as cosine similarity. We hypothesize that there is a shared challenge in using frozen features directly for the "masked signal modeling" pre-training paradigm. Some previous works in NLP attribute this to the anisotropic problem (Ethayarajh, 2019; Li et al., 2020): the representations are not uniformly distributed with respect to direction and only occupy a narrow cone in the embedding space. To achieve good VRF performance, we expect visual representations can be utilized directly without further compute-intensive fine-tuning.

5. Conclusion and Discussion

The proliferation of pre-trained vision models for visuomotor control is an exciting and ongoing event. However, there has been a noticeable lack of work on evaluation protocols in this area. We conduct the first thorough empirical evaluation of pre-trained vision model performance across different downstream policy learning methods and environments.

Our evaluation shows the following insights: (*i*) The effectiveness of a pre-trained vision model highly depends on the downstream policy learning methods. The vision of a '*universal*' pre-trained model with the best performance on all control tasks is yet to be realized. (*ii*) Due to randomness that cannot be mitigated, RL methods demonstrate overly high variability, and thus cannot be positioned as reliable evaluation methods for vision models. (*iii*) Without control-specific adaptation, BC still benefits from the latest benchmark-leading models in the vision community, due to their inherent ability to capture more environment-relevant information such as object locations and joint positions. (*iv*) Different vision models yield the most consistent performance when using VRF, which requires the vision model to learn global features and capture a notion of task progress. MAE is a noticeable underperforming outlier, likely due to the fact that it suffers from the anisotropic problem.

Our findings highlight the importance of evaluating vision models for motor control in a comprehensive manner. We

advocate for using more consistent and robust evaluation methods such as VRF and BC to minimize uncertainty and obtain reliable results, as well as claim the effectiveness of the pre-trained vision models in terms of which specific downstream policy learning algorithms are targeted. Although we are unable to provide guidance to researchers who rely on RL due to the unavailability of expert demonstrations, we believe that this limitation is temporary. Our work, as well as other related studies, demonstrate the high variability of RL, which could motivate researchers to develop low-variance RL algorithms. Once such an algorithm is available, it will be possible to evaluate different pre-trained vision models using RL.

With the rapid development of pre-trained vision models for motor control, we believe there is an urgent need to benchmark their empirical performance. To further support researchers in this endeavor, we will release a library including our evaluation benchmark and pre-trained models. We hope our work will help measure progress in this field and provide common ground for future comparisons.

Acknowledgements

This work is supported by the Ministry of Science and Technology of the People's Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000). This work is also supported by the National Key R&D Program of China (2022ZD0161700).

References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Bardes, A., Ponce, J., and LeCun, Y. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Chan, S. C., Fishman, S., Canny, J., Korattikara, A., and Guadarrama, S. Measuring the reliability of reinforcement learning algorithms. *arXiv preprint arXiv:1912.05663*, 2019.

Chen, A. S., Nair, S., and Finn, C. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021a.

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

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021b.

Cohen, S., Amos, B., Deisenroth, M. P., Henaff, M., Vinitsky, E., and Yarats, D. Imitation learning from pixel observations for continuous control. 2021.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–736, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dong, X., Zheng, Y., Bao, J., Zhang, T., Chen, D., Yang, H., Zeng, M., Zhang, W., Yuan, L., Chen, D., et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. *arXiv preprint arXiv:2208.12262*, 2022.

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

Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., 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*, pp. 18995–19012, 2022.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Haldar, S., Mathur, V., Yarats, D., and Pinto, L. Watch and match: Supercharging imitation with regularized optimal transport. *arXiv preprint arXiv:2206.15469*, 2022.

Hansen, N., Yuan, Z., Ze, Y., Mu, T., Rajeswaran, A., Su, H., Xu, H., and Wang, X. On pre-training for visuomotor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

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

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

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

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Hu, Y., Wang, R., Zhang, K., and Gao, Y. Semantic-aware fine-grained correspondence. In *European Conference on Computer Vision*, pp. 97–115. Springer, 2022.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Khatib, O. Inertial properties in robotic manipulation: An object-level framework. *The international journal of robotics research*, 14(1):19–36, 1995.

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

Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., and Tompson, J. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Kumar, S., Zamora, J., Hansen, N., Jangir, R., and Wang, X. Graph inverse reinforcement learning from diverse videos. *Conference on Robot Learning (CoRL)*, 2022.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895, 2020a.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020b.

LeCun, Y. A path towards autonomous machine intelligence. *preprint posted on openreview*, 2022.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

Mu, N., Kirillov, A., Wagner, D., and Xie, S. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pp. 529–544. Springer, 2022.

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

Ng, A. Y., Russell, S., et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

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

Papagiannis, G. and Li, Y. Imitation learning with sinkhorn distances. *arXiv preprint arXiv:2008.09167*, 2020.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

Peng, Z., Dong, L., Bao, H., Ye, Q., and Wei, F. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

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

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., and Darrell, T. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Shah, R. and Kumar, V. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

Shan, D., Geng, J., Shu, M., and Fouhey, D. F. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9869–9878, 2020.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. PMLR, 2014.

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

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

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

Wang, C., Luo, X., Ross, K., and Li, D. Vrl3: A data-driven framework for visual deep reinforcement learning. *arXiv preprint arXiv:2202.10324*, 2022.

Wang, X., Zhang, R., Shen, C., Kong, T., and Li, L. Dense contrastive learning for self-supervised visual pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3024–3033, 2021.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14668–14678, 2022.

Wen, C., Lin, J., Darrell, T., Jayaraman, D., and Gao, Y. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 33:2564–2575, 2020.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

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

Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., and Hu, H. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16684–16693, 2021.

Xu, J. and Wang, X. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10075–10085, 2021.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10674–10681, 2021b.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., and Xu, H. Pre-trained image encoder for generalizable visual reinforcement learning. *arXiv preprint arXiv:2212.08860*, 2022.

Zakka, K., Zeng, A., Florence, P., Tompson, J., Bohg, J., and Dwibedi, D. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pp. 537–546. PMLR, 2022.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Zhu, Y., Wong, J., Mandlekar, A., and Martín-Martín, R. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

A. Policy Learning Methods

A.1. Reinforcement Learning: DrQ-v2

We employ DrQ-v2 (Yarats et al., 2021a), an off-policy actor-critic approach for continuous vision-based control, as our reinforcement learning algorithm. The original formulation of DrQ-v2 relies on data augmentations (e.g., random shifts) (Kostrikov et al., 2020; Laskin et al., 2020a) to facilitate the learning of a shallow *randomly initialized* image encoder. Our *frozen* encoder network f is pre-trained on real-world images, and the representations already possess some general knowledge about the environment. Thus, we do not apply any data augmentation to the image observation. This makes it possible to store representation vectors $z_t = f(o_t)$ directly in the replay buffer instead of raw image observations. The actor and critic networks can take in the representations sampled from the replay buffer, without re-encoding via the encoder, significantly speeding up the training process.

The core of DrQ-v2 is Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) augmented with n-step returns. The critic is trained using clipped double Q-learning (Fujimoto et al., 2018) to reduce the overestimation bias in the target value. Specifically, this requires training two Q-functions Q_{θ_1} and Q_{θ_2}. The critic loss for each Q-function is given by:

$$\mathcal{L}_{\theta_k}(\mathcal{D}) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\left(Q_{\theta_k}(z_t, a_t) - y\right)^2\right] \quad \forall k \in \{1, 2\} \tag{1}$$

where τ denotes a mini-batch of transitions $(z_t, a_t, r_{t:t+n-1}, z_{t+n})$ sampled from the replay buffer $\mathcal{D}$ and y is the TD target defined as:

$$y = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n \min_{k=1,2} Q_{\bar{\theta}_k}(z_{t+n}, a_{t+n}) \tag{2}$$

Here, $a_{t+n} = \pi_\phi(z_{t+n}) + \epsilon$ and $\bar{\theta}_1, \bar{\theta}_2$ are the slow-moving weights of target Q-networks. The exploration noise ϵ is sampled from $\text{clip}\left(\mathcal{N}\left(0, \sigma^2\right), -c, c\right)$, with variance σ^2 following a linear decay schedule. Finally, the deterministic actor π_ϕ is trained using deterministic policy gradients (DPG) (Silver et al., 2014) with the following loss:

$$\mathcal{L}_\phi(\mathcal{D}) = -\mathbb{E}_{z_t\sim\mathcal{D}}\left[\min_{k=1,2} Q_{\theta_k}(z_t, a_t)\right] \tag{3}$$

where $a_t = \pi_\phi(z_t) + \epsilon$, and $\epsilon \sim \text{clip}\left(\mathcal{N}\left(0, \sigma^2\right), -c, c\right)$.

A.2. Imitation Learning with Visual Reward Functions: ROT

We adopt Regularized Optimal Transport (ROT) (Haldar et al., 2022) as one representative algorithm for imitation learning with a visual reward function. ROT derives the imitation reward r^e based on the Sinkhorn distance between the agent and the expert observations. In particular, ROT interprets an observation trajectory $(o_1, \ldots, o_T)$ as a discrete probability measure of the form $\mu_o = \frac{1}{T}\sum_{t=1}^{T} \delta_{o_t}$. The closeness between agent trajectories $o^a_{1:T}$ and expert trajectories $o^e_{1:T}$ can be computed by measuring the optimal transport of probability mass from $o^a_{1:T} \rightarrow o^e_{1:T}$. We encode trajectories using *frozen* pre-trained vision models:

$$z^a = \left[f(o^a_1), \ldots, f(o^a_T)\right] \quad z^e = \left[f(o^e_1), \ldots, f(o^e_T)\right] \tag{4}$$

Given a cosine cost matrix computed between encoded image observations $C_{t,t'} = 1 - \frac{\langle z^a_t, z^e_{t'}\rangle}{\|z^a_t\|\cdot\|z^e_{t'}\|}$, the optimal alignment between an encoded expert trajectory z^e and an encoded agent trajectory z^a can be computed as:

$$\mu^* \in \operatorname*{argmin}_{\mu\in\mathcal{M}} \sum_{t,t'=1}^{T} C_{t,t'}\mu_{t,t'} \tag{5}$$

where $\mathcal{M} = \left\{\mu \in \mathbb{R}^{T\times T} : \mu\mathbf{1} = \mu^T\mathbf{1} = \frac{1}{T}\mathbf{1}\right\}$ is the set of coupling matrices. Since solving Eq. 5 is computationally expensive, ROT uses the Sinkhorn algorithm to get approximate solutions quickly (Cuturi, 2013; Papagiannis & Li, 2020; Cohen et al., 2021).

Finally, the imitation reward for each agent observation can be extracted using the equation:

$$r^e\left(o_t^a\right) = -\sum_{t'=1}^{T} C_{t,t'}\mu^*_{t,t'} \tag{6}$$

Given the computed imitation reward, ROT employs DrQ-v2 as the underlying RL optimizer to achieve efficient reward maximization. Intuitively, this encourages the agent to generate trajectories that closely match demonstrated expert trajectories.

ROT combines the inverse reinforcement learning (IRL) process with behavior cloning (BC) to further improve sample efficiency and final performance. This is done in two phases. In the first phase, BC is used to pre-trained a randomly initialized policy. In the second phase, the pre-trained policy is fine-tuned with the IRL objective and BC objective simultaneously, where the BC loss is added to the IRL objective with an adaptive weight. Strictly speaking, ROT is not a pure IRL algorithm. However, the primary role of BC is to 'regularize' IRL, and we empirically find that the quality of the imitation reward is the decisive factor. This is evidenced by the poor performance of MAE (see Figure 8). Despite the assistance of BC, the scores achieved by MAE on most tasks are still *zero*. We believe that using ROT as a downstream policy learning method can reflect the suitability of different pre-trained vision models as good visual reward functions.

B. Pre-Trained Vision Models

Throughout the paper, we consider 14 pre-trained vision models covering prevalent pre-training methods. Here, we give a brief description of each of the models.

MoCo v2 (He et al., 2020; Chen et al., 2020b) is a classic unsupervised visual representation learning method. MoCo v2 builds upon the instance discrimination task (Wu et al., 2018) that considers each image of the dataset (or "instance") and its transformations as a separate class. In addition, MoCo v2 uses an explicit momentum encoder to build large and consistent negative samples for contrastive learning. The representations learned transfer well and match the performance of the supervised pre-training counterpart. PVR (Parisi et al., 2022) finds that MoCo v2 representations can be competitive or even better than hand-engineered ground-truth features to train motor control policies. We note, that in contrast to PVR, we do not use representations from multiple layers and only use the representations from `res5` block (last block) for fair comparisons with other ResNet models.

SwAV (Caron et al., 2020) is a clustering-based method (Caron et al., 2018) for unsupervised visual representation learning. SwAV forces the representations of different images to belong to different clusters on the unit sphere, which is achieved by computing the assignment from one image view and predicting it from another image view. SwAV performs online clustering under a balanced partition constraint for each batch, which ensures that the assignment to clusters is as uniform as possible.

SimSiam (Chen & He, 2021) uses simple Siamese networks (Bromley et al., 1993) to learn meaningful representations by directly maximizing the similarity of one image's two views. Neither negative pairs nor a momentum encoder is used. The stop-gradient operation in SimSiam plays an essential role in avoiding collapsing solutions.

DenseCL (Wang et al., 2021), short for dense contrastive learning, is a self-supervised learning method that operates directly at the levels of local features. DenseCL defines the positive sample of each local feature vector by extracting the correspondence across two views and optimizing a pairwise contrastive (dis)similarity loss at the local feature level. The learned representations preserve spatial information, which is beneficial for dense prediction tasks like semantic segmentation and object detection.

PixPro (Xie et al., 2021) is similar to DenseCL in that it utilizes dense pretext tasks for self-supervised visual representation learning. But PixPro uses a BYOL-style (i.e., non-contrastive) (Grill et al., 2020) training framework, eliminating the need for negative samples. Specifically, PixPro learns the representations by pulling local feature vectors belonging to the same spatial region close together.

VICRegL (Bardes et al., 2022) combines the best of global feature learning and local feature learning. This is achieved by applying the VICReg criterion (Bardes et al., 2021) to pairs of global feature vectors and pairs of local feature vectors simultaneously. Two local feature vectors are designated as positive pairs and attracted to each other if their ℓ_2 distance is below a threshold or if their relative locations are consistent with a known geometric transformation between the two input views. VICRegL achieves excellent performance on detection and segmentation tasks while maintaining good performance on classification tasks.

VFS (Xu & Wang, 2021) is self-supervised pre-trained on a large-scale video dataset, Kinetics400 (Kay et al., 2017), while all the other models above are pre-trained on a static image dataset, ImageNet (Russakovsky et al., 2015). The pre-training pipeline of VFS is similar to that of SimSiam, with the exception that VFS considers frames at different video timestamps as different views for similarity learning. We use VFS to investigate if representations that capture temporal information are beneficial for control policy learning.

R3M (Nair et al., 2022) is a vision model designed to enable data-efficient behavior cloning for robotic manipulation tasks. R3M is pre-trained on the large-scale Ego4D human video dataset (Grauman et al., 2022). R3M leverages time-contrastive learning (Sermanet et al., 2018), video-language alignment, and an L1 penalty to encourage sparse and compact representations. R3M represents a significant advancement in the field of pre-trained vision models for motor control, and should be taken into account in our large-scale benchmarking studies.

VIP (Ma et al., 2022) is designed to provide visual representations and dense reward signals for robotic manipulation tasks. VIP treats representation learning as an offline goal-conditioned reinforcement learning problem and solves the Fenchel dual problem of goal-conditioned value function learning that does not depend on actions, enabling pre-training on large-scale human video datasets.

MoCo v3 (Chen et al., 2021b) is a straightforward extension of MoCo v2 by replacing the backbone architecture from ResNet (He et al., 2016) to Vision Transformers (ViT) (Dosovitskiy et al., 2020). However, instability is a major issue in self-supervised ViT training. MoCo v3 alleviates the instability issue by simply freezing the patch projection layer in ViT.

DINO (Caron et al., 2021) is a simple approach for self-supervised ViT training that can be interpreted as a form of knowledge distillation (Hinton et al., 2015) without labels. Specifically, DINO uses a student network to predict the output of a momentum-updated teacher network with a standard cross-entropy loss. To avoid collapse solutions, a centering and sharpening operation is applied to the momentum teacher outputs. The resulting features of DINO are biased towards shape and explicitly contain the scene layout information of an image.

MAE (He et al., 2022) is pre-trained by the masked image modeling (MIM) task (Bao et al., 2021) inspired by the success of masked language modeling in NLP (Devlin et al., 2018). MAE adopts an asymmetric encoder-decoder architecture. The encoder is only applied to the visible subset of patches (without mask tokens). The lightweight decoder reconstructs the image from the encoded visible patches and mask tokens. The mask ratio is as high as 75% to reduce the heavy spatial redundancy of images. MVP (Xiao et al., 2022) uses an MAE model pre-trained on images in the wild, e.g., from YouTube (Shan et al., 2020) or Egocentric videos (Damen et al., 2018) to provide effective representations for motor control. In this study, we adopt the original MAE pre-trained on ImageNet (Russakovsky et al., 2015) for fair comparisons with other models.

iBOT (Zhou et al., 2021) can be viewed as the combination of self-distillation (DINO) and masked image modeling. Self-distillation is used to train a teacher network/online tokenizer that captures high-level visual semantics. Masked image modeling is used to train a student network that can recover each masked patch token to its corresponding teacher network/online tokenizer output.

CLIP (Radford et al., 2021) is a simple and scalable method to learn visual representations with language supervision. CLIP forces the image representations to be aligned with paired captions through contrastive learning. The learned representations have strong zero-shot transferability and are effective for some robotic manipulation (Shridhar et al., 2022) and Embodied AI tasks (Khandelwal et al., 2022).

C. Environments

We use 21 tasks across 3 robot manipulation environments: Meta-World (8 tasks), Robosuite (8 tasks), and Franka-Kitchen (5 tasks). We compare these 3 environments to other commonly used environments in Table C.1.

Table C.1: Comparisons of different control environments. *Kitchen* is short for Franka-Kitchen. Habitat does not support full-physics simulation and is only suitable for studying high-level planning. DeepMind Control (DMC) Suite focuses on locomotion tasks and the observations are not visually realistic. RLbench (James et al., 2020) has the most tasks but sparse rewards and low simulation speed.

	Meta-World	Robosuite	Kitchen	Habitat	DMC	RLbench
Full-physics	✓	✓	✓	✗	✓	✓
Visually-realistic	✓	✓	✓	✓	✗	✓
Dense rewards	✓	✓	✓	✓	✓	✗
Simulator	MuJoCo	MuJoCo	MuJoCo	Habitat-Sim	MuJoCo	Coppelia

Meta-World (Yu et al., 2020) is a simulated environment for multi-task learning and meta-learning consisting of 50 robotic manipulation tasks. We select 8 distinct tasks: `hammer`, `drawer-close`, `door-open`, `bin-picking`, `button-press-topdown`, `window-close`, `lever-pull` and `coffee-pull`, depicted in Figure C.1. All the tasks are performed by a simulated Sawyer robot. The episode length for each task is 125 steps, except for `bin-picking` and `lever-pull`, which run for 175 steps. The positions of the target objects (e.g., hammer, drawer, door, etc.) are randomized between episodes. For each task, we use the task-specific hard-coded policies provided in the open-source implementation to collect a total of 25 expert demonstrations. Our BC experiments consider a varying number of expert demonstrations: [1, 5, 10, 15, 20, 25], but for all the VRF experiments, we only use 1 demonstration.

Robosuite (Zhu et al., 2020) is a modular simulation framework and benchmark for robot learning. We consider 8 tasks from the environment, including `Panda-Lift`, `Panda-Door`, `Panda-TwoArmPegInHole`, `Panda-PickPlaceCan`, `Panda-NutAssemblySquare`, `Jaco-Lift`, `Jaco-Door` and `Jaco-TwoArmPegInHole`, as illustrated in Figure C.2. 5 out of 8 tasks are performed by a Panda robot with a parallel-jaw gripper, while the other 3 tasks are performed by a Jaco robot with multi-jointed fingers, which are harder to control. We use Operational Space Controllers (OSC) (Khatib, 1995) to transform the high-level actions into low-level virtual motor commands. The episode length for all tasks is 80 steps (except `Panda-PickPlaceCan` and `Panda-NutAssemblySquare`, which runs for 150 and 160 steps, respectively). In all tasks, the locations of objects (e.g., cube, can, nut, etc.) are randomized at the beginning of each episode. For `Panda-PickPlaceCan` and `Panda-NutAssemblySquare`, we use 50 expert demonstrations provided by robomimic (Mandlekar et al., 2021). For the other tasks, we train a state-based DrQ-v2 and collect 50 demonstrations for each task. Our BC experiments consider a different number of expert demonstrations: [5, 10, 20, 30, 40, 50], but all the VRF experiments only use 1 demonstration.

Franka-Kitchen (Gupta et al., 2019) requires to control a 9 DoF Franka robot to perform various tasks in a kitchen scene. In this study, we consider 5 tasks: `knob1-on`, `micro-open`, `ldoor-open`, `light-on` and `sdoor-open`, as depicted in Figure C.3. The episode length for all Franka tasks is 50 steps. Following PVR (Parisi et al., 2022), we randomize the pose of the robot arm between episodes but not the scene itself. We train expert policies using state-based DrQ-v2 and collect 25 expert demonstrations for each task. Our BC experiments consider a varying number of expert demonstrations: [1, 5, 10, 15, 20, 25], but for all VRF experiments, we only use 1 demonstration.

D. Implementation Details

D.1. Policy Learning

For all the environments and tasks, the observations are 224×224 RGB images with no access to proprioceptive information. In our experiments, we find that using only one image observation is comparable to using a stack of consecutive images, so we choose the more compute-efficient option. Additionally, we do not use image augmentation (e.g., random shift). For all Meta-World and Robosuite tasks, we employ action repeat of 2, while for Franka-Kitchen tasks, the action repeat is set to 1 (no action repeat). Further implementation details for each policy learning method are as follows.

Reinforcement learning. All the policies are trained for 3M environment steps. We set the replay buffer size to 500000 and increase the mini-batch size to 512. The features output by a pre-trained vision model are fed into the actor and critic networks, whose architectures follow DrQ-v2: a 'trunk' network and a small MLP network. The 'trunk' network is a single fully-connected layer with `LayerNorm` (Ba et al., 2016) and `tanh` nonlinearity. The output dimension of the 'trunk' network is 50, forming a bottleneck structure. The MLP network has 3 layers, and the hidden dimension is set to 1024. The

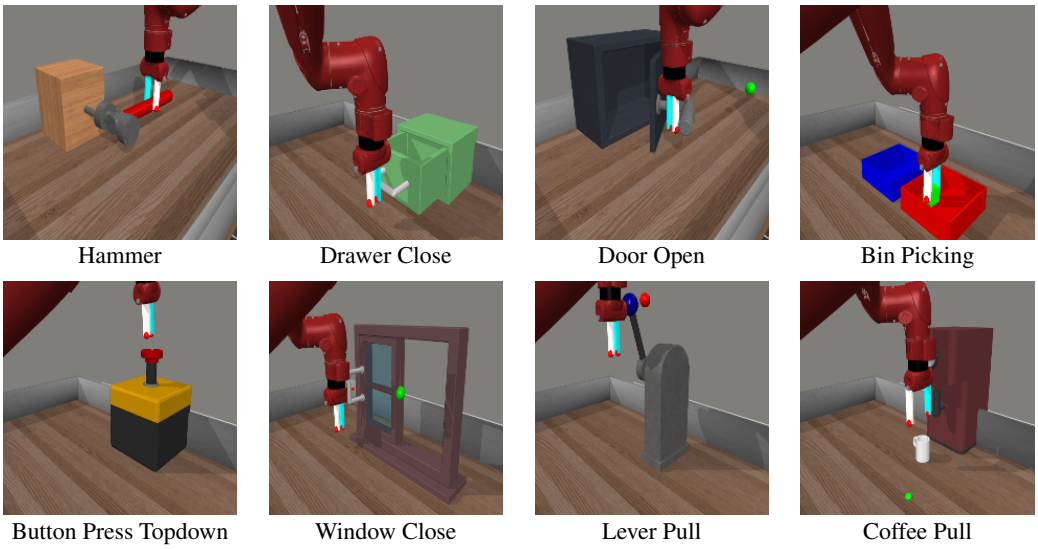


Figure C.1: 8 Meta-World tasks we consider in our study.

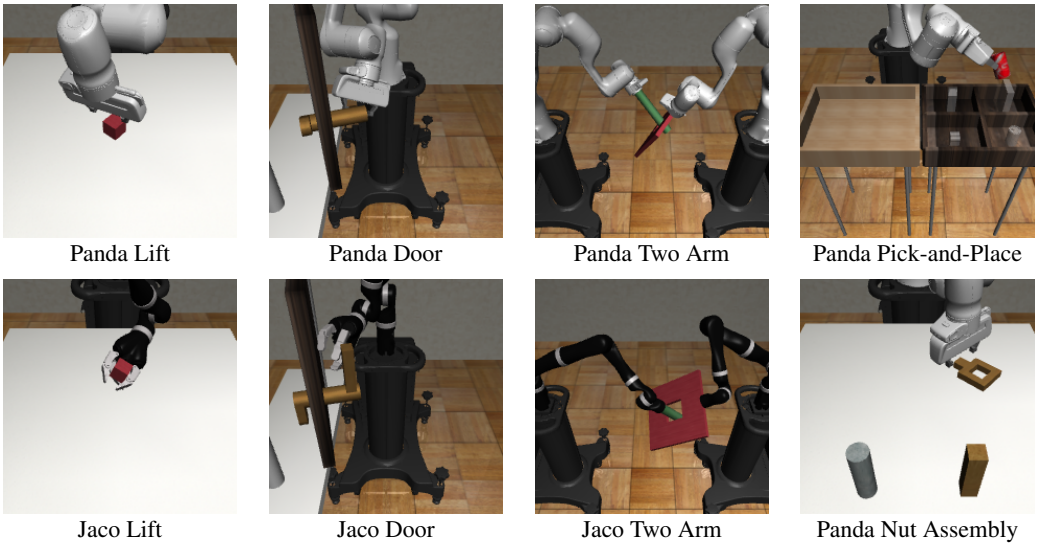


Figure C.2: 8 Robosuite tasks we consider in our study.

Figure C.3: 5 Franka-Kitchen tasks we consider in our study.

complete list of hyper-parameters is in Table D.1.

Imitation learning through behavior cloning. For BC, we discard the bottleneck structure in DrQ-v2's actor and use an architecture as follows: a four-layer MLP with hidden sizes [512, 256, 128] and `ReLU` activations. The MLP network is

preceded by a `LayerNorm` layer to calibrate the feature magnitudes across different pre-trained vision models. We train the policy with mini-batches of 128 samples for 110000 steps with the Adam optimizer (Kingma & Ba, 2014) (learning rate 0.0001).

Imitation learning with a visual reward function. We closely follow the implementation of ROT. All the policies are trained for 1M environment steps. As we can compute the imitation rewards on the output features of pre-trained vision models, we no longer need the target feature processor in the original ROT. See Table D.2 for all the hyper-parameters.

Table D.1: A default set of hyper-parameters for RL.

Config	Value
Training environment steps	3.1×10^6
Replay buffer capacity	500000
Seed frames	4000
Exploration steps	2000
n-step returns	3
Mini-batch size	512
Discount γ	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Features dim.	50
Hidden dim.	1024
Exploration stddev. clip	0.3
Exploration stddev. schedule	$\text{linear}(1.0, 0.1, 800000)$

Table D.2: A default set of hyper-parameters for VRF.

Config	Value
Training environment steps	1.1×10^6
Replay buffer capacity	150000
Seed frames	12000
Exploration steps	0
n-step returns	3
Mini-batch size	256
Discount γ	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Features dim.	50
Hidden dim.	1024
Exploration stddev. clip	0.3
Exploration stddev. schedule	0.1
BC weight type	qfilter
Auto reward scale factor	10

D.2. Linear Probing

We train a linear regressor on Franka-Kitchen image features output by frozen pre-trained vision models to predict the ground-truth environment features, which are composed of joint positions, object locations, end-effector poses, and relative distances between object and end-effector. The dimension of this environment feature is 57. We adopt an extra `LayerNorm`

layer before the linear regressor to calibrate the feature magnitudes across different vision models. We use $lr = 0.02$ with a warmup of 10 epochs and cosine learning rate decay, weight decay $= 0$, and batch size $= 16$ with an Adam optimizer. We train on 40 expert demonstrations (2000 images) for 100 epochs and report validation loss on 10 expert demonstrations.

E. Additional Experimental Results

E.1. Reinforcement Learning

Folk wisdom in experimental RL suggests that evaluating more runs per task can reduce uncertainty. In this section, we investigate the following question: "How many runs with different seeds are required to obtain stable and reliable results?" By observing the 95% confidence intervals in Figure E.1, we find that there is substantial uncertainty in scores even with 50 runs and it may be necessary to perform 100 or more runs to address the issue of statistical uncertainty effectively.

Next, we ask the question: "How many training runs do we need to compare two pre-trained vision models reliably?" We show the *probability of improvement* (Agarwal et al., 2021) of R3M compared to VFS for a varying number of runs in Figure E.2. With a handful of runs (i.e., 3 or 5), the confidence intervals are substantially large. The comparison is stable and reliable only when 40 or more runs are used, however, this number of runs is often computationally prohibitive for complex and challenging benchmarks.

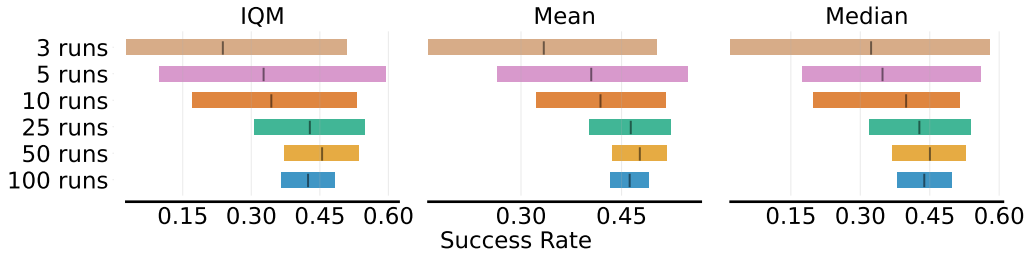


Figure E.1: RL performance with 95% CIs for *a varying number of runs* for IQM, Mean and Median scores for R3M. We use 5 tasks from Meta-World: `hammer`, `door-open`, `bin-picking`, `button-press-topdown`, `lever-pull` and train all the RL agents for 2M environment steps. The other experimental settings are the same as described in Appendix D.1.

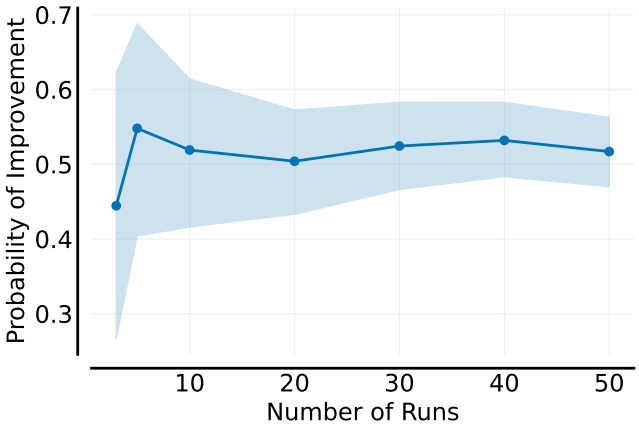


Figure E.2: The probability of improvement, with 95% CIs (shaded regions), that R3M outperforms VFS, for *a varying number of runs*. The interval estimates are based on stratified bootstrap with independent sampling with 2000 bootstrap re-samples. We consider 5 tasks (the same as Fig. E.1) from Meta-World and train all the RL agents for 2M environment steps. The other settings are the same as described in Appendix D.1.

E.2. Full Aggregate Metrics on BC and VRF

Figure E.3, Figure E.4, Figure E.5 and Figure E.6 show the BC results across 3 aggregate metrics (IQM, Mean, Median) on Meta-World, Robosuite, Franka-Kitchen and all environements, respectively.

Figure E.7, Figure E.8, Figure E.9 and Figure E.10 show the VRF results across 3 aggregate metrics (IQM, Mean, Median) on Meta-World, Robosuite, Franka-Kitchen and all environements, respectively.

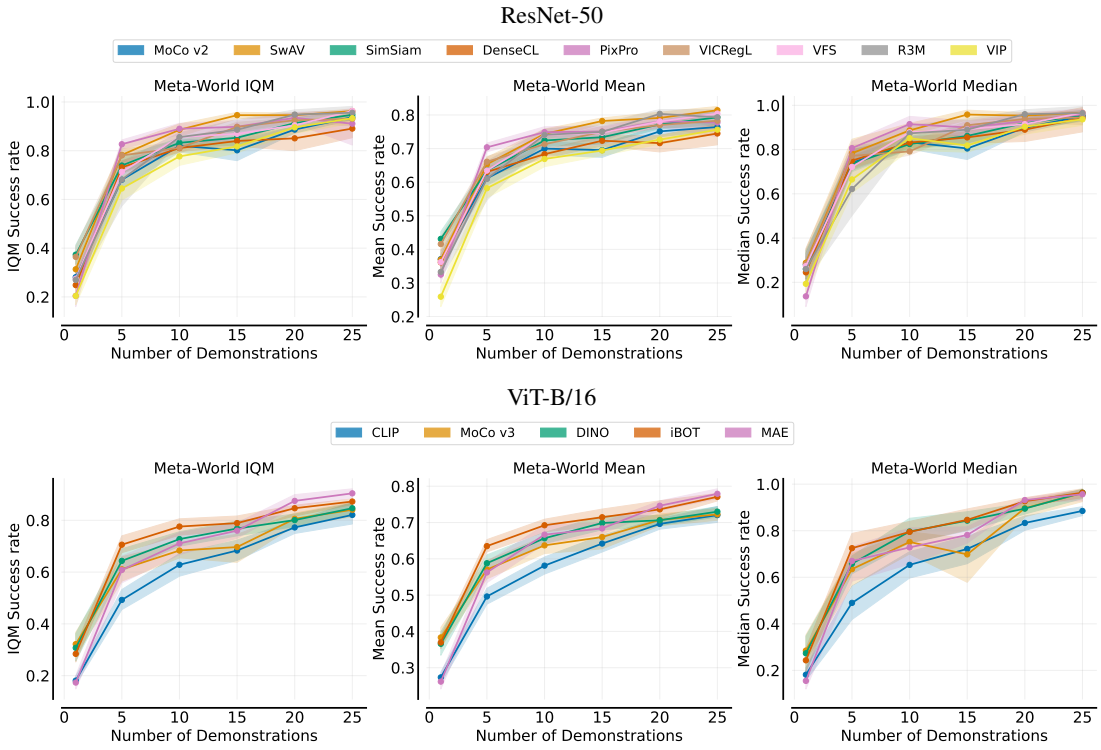


Figure E.3: BC performance across 3 aggregate metrics (IQM, Mean, and Median) on Meta-World. Shaded regions show pointwise 95% percentile stratified bootstrap CIs.

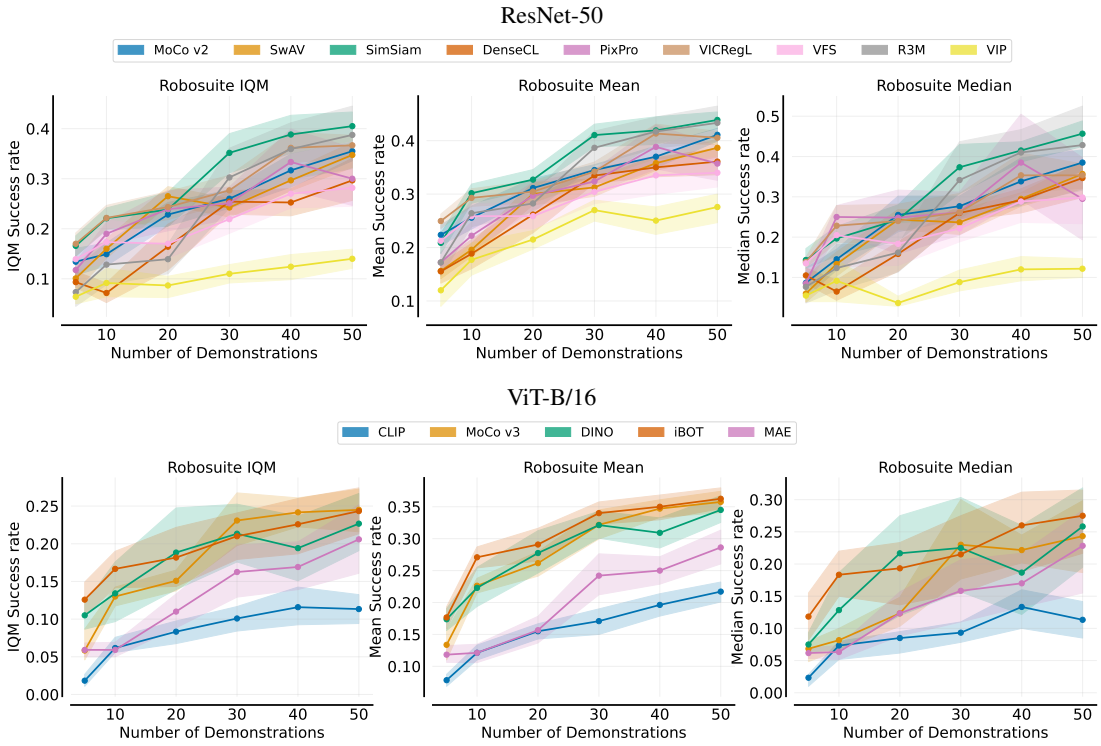


Figure E.4: BC performance across 3 aggregate metrics (IQM, Mean, and Median) on Robosuite. Shaded regions show pointwise 95% percentile stratified bootstrap CIs.

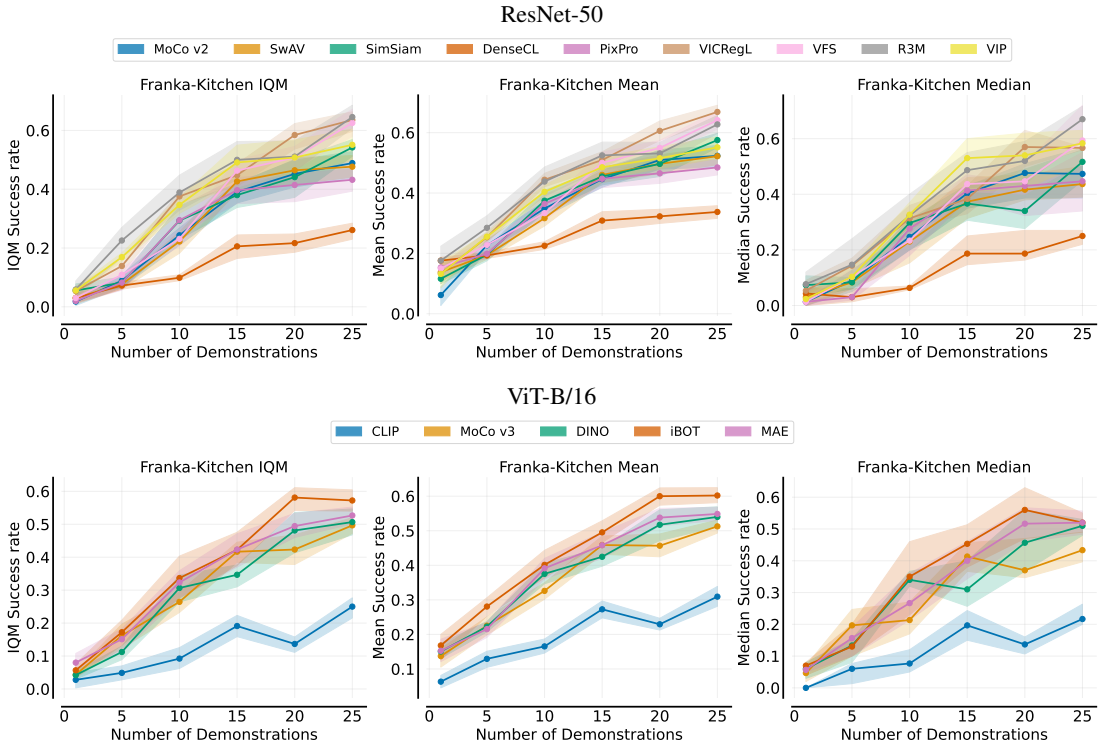


Figure E.5: BC performance across 3 aggregate metrics (IQM, Mean, and Median) on Franka-Kitchen. Shaded regions show pointwise 95% percentile stratified bootstrap CIs.

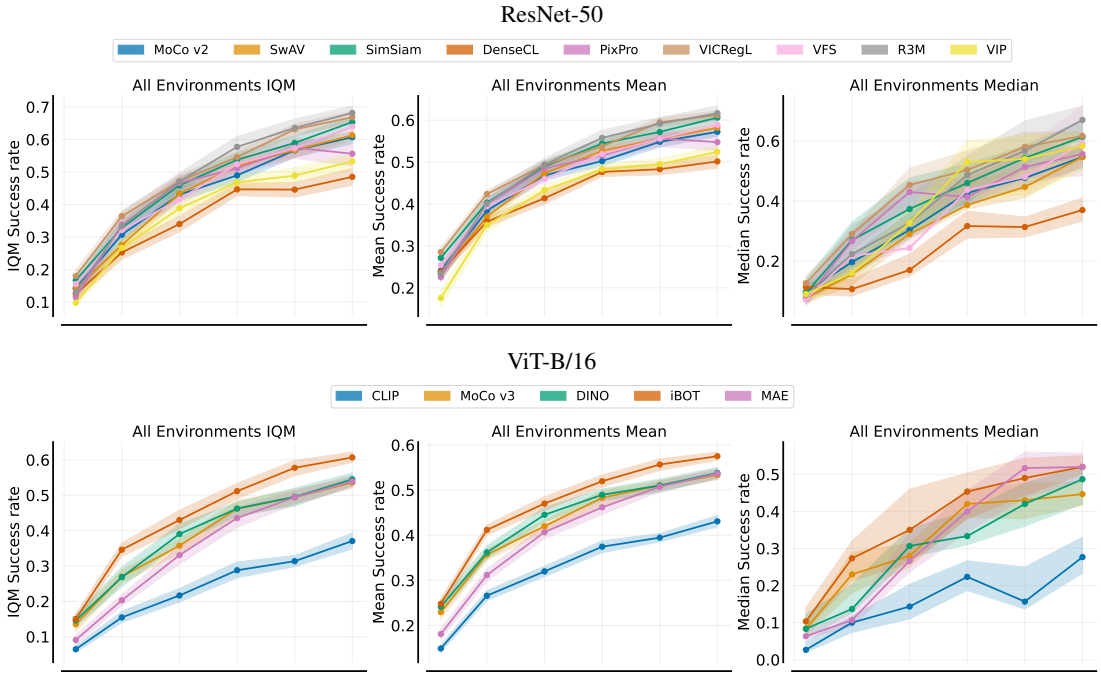


Figure E.6: BC performance across 3 aggregate metrics (IQM, Mean, and Median) on 21 tasks of all environments. Shaded regions show pointwise 95% percentile stratified bootstrap CIs.

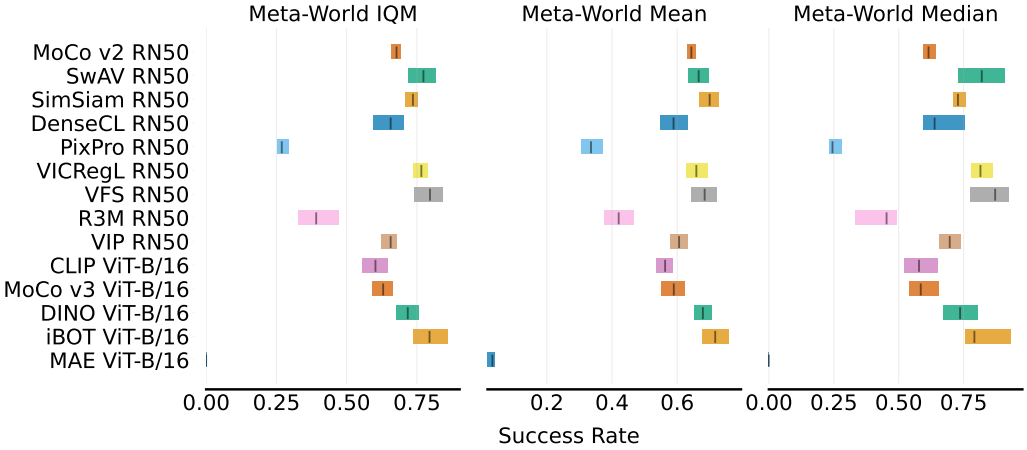


Figure E.7: VRF performance across 3 aggregate metrics (IQM, Mean, and Median) on Meta-World with 95% CIs.

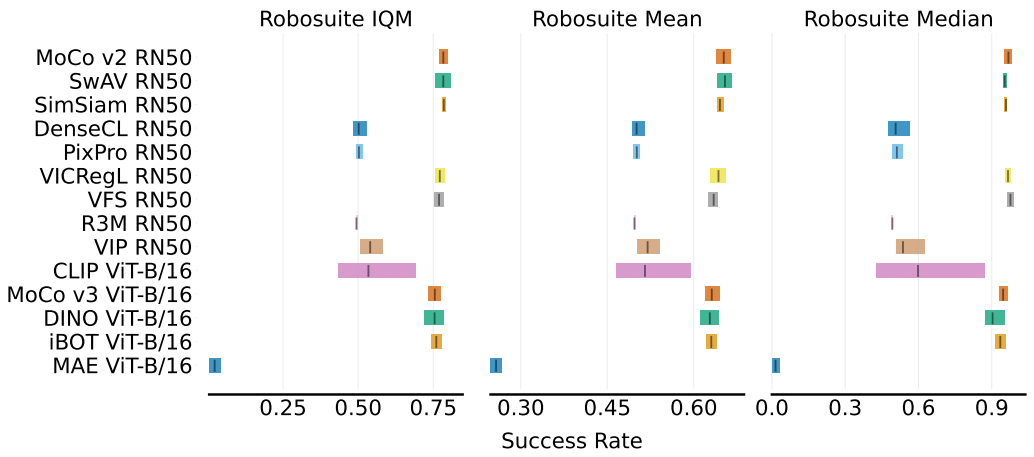


Figure E.8: VRF performance across 3 aggregate metrics (IQM, Mean, and Median) on Robosuite with 95% CIs..

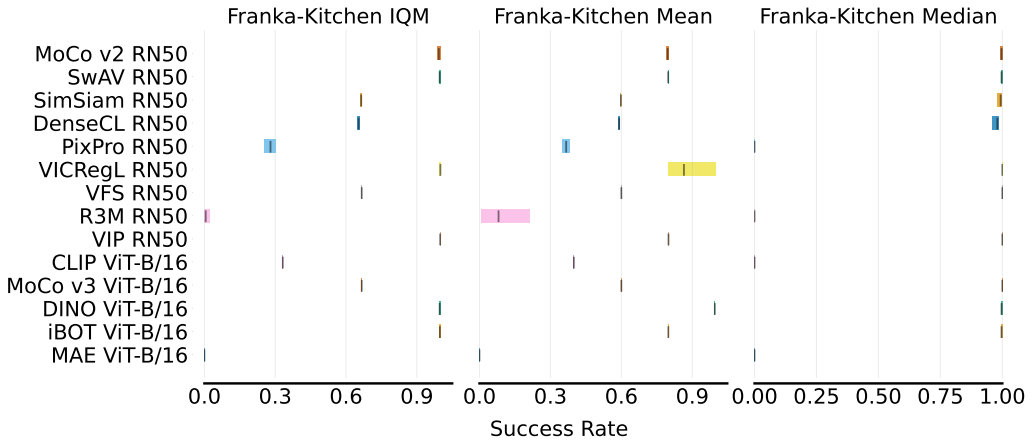


Figure E.9: VRF performance across 3 aggregate metrics (IQM, Mean, and Median) on Franka-Kitchen with 95% CIs.

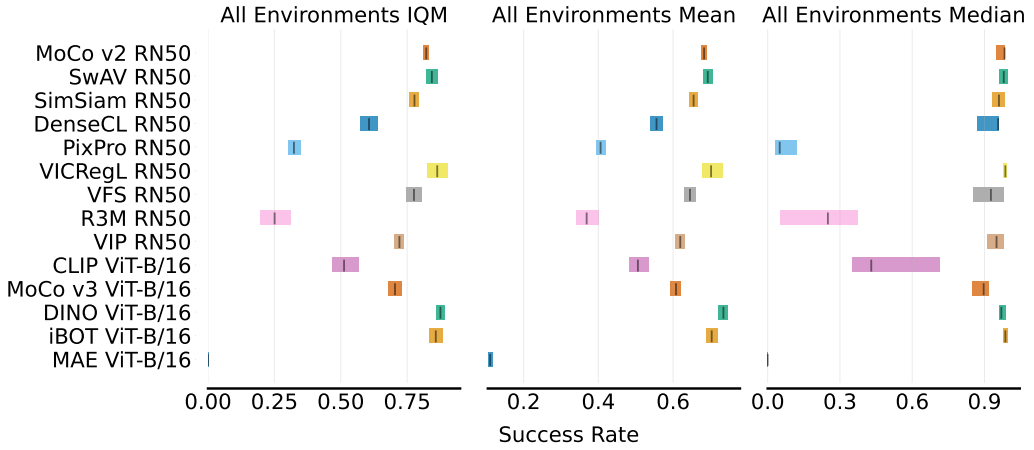


Figure E.10: VRF performance across 3 aggregate metrics (IQM, Mean, and Median) on 21 tasks of all environments with 95% CIs.