Robot Learning with Sensorimotor Pre-training

Anonymous Author(s)
Affiliation
Address
email

Abstract: We explore self-supervised sensorimotor pre-training for robotics. We formulate pre-training as a sensorimotor prediction problem. Given a sequence of camera images, proprioceptive robot states, and past robot actions, we mask out a random subset of the sequence and train a model to predict the missing content. We hypothesize that if the robot can predict the masked-out content it has acquired a good representation of the physical world that can enable it to act. Our Robotic Pre-trained Transformer, which we call RPT, is designed to work on latent visual representations which enables scaling to $10\times$ larger models, fast training, and real-time inference on a physical robot. To study our pre-training approach, we collect a dataset of over 20,000 real-world trajectories over 9 months using a combination of motion planning and model-based grasping algorithms. We find that pre-training on this data consistently outperforms training from scratch, leads to $2\times$ improvements in the block stacking task, and has favorable scaling properties.

Keywords: Self-supervised, Sensorimotor, Pre-training, Sequence Modeling

1 Introduction

Over the last few year there has been an increased interest in pre-training for robotics. For example, we have seen promising results in pre-training visual representations from human videos [1, 2]. More recently, we have seen strong results from training visuomotor models on a large dataset of robotic trajectories annotated with language [3]. We ask: *can we learn good representations for robotics from sensorimotor trajectories alone?*

In this paper, we propose a self-supervised sensorimotor pre-training approach for robotics. We formulate pre-training via a sensorimotor prediction problem. We hypothesize that if the robot can predict the missing content it has learned something about the physical world that can enable it to act. We instantiate this idea through a masked reconstruction task, similar to the BERT [4] and MAE [5] counterparts in language and computer vision.

Our model, called RPT, is a Transformer [6] that operates on sequences of sensorimotor tokens. Specifically, given an input sequence of interleaved camera images, proprioceptive states, and actions, we mask out a random subset of the sequence, and predict the missing content from the rest. We perform masking uniformly at random across all modalities and time using a high masking ratio, which encourages the model to learn cross-modal, spatio-temporal representations.

We encode camera images using a pre-trained vision encoder [1] and use latent visual representations for sensorimotor sequence learning. This enables us to benefit from diverse visual data from the Internet that does not come with the accompanying proprioceptive and action information. Moreover, this design decouples the vision cost from the sensorimotor context length, making real-time control with over 300M parameter models and large context lengths feasible.

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

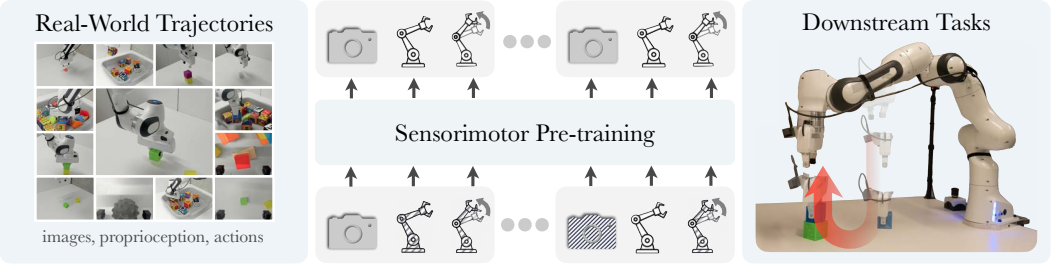


Figure 1: **Framework.** *Left:* We collect a large dataset of real robot trajectories that contains multiview RGB images, proprioceptive robot states, and actions for self-supervised learning. *Middle:* Robotic Pre-trained Transformer (RPT) is trained to reconstruct masked multimodal inputs (marked by the striped pattern). *Right:* We evaluate the model on real-world robotics tasks, such as cube stacking (shown), demonstrating the effectiveness of pre-training in reducing sample complexity.

To study our pre-training approach, we collect a dataset of over 20,000 real-world trajectories using a combination of motion planning and model-based grasping algorithms. Each trajectory is a sequence of multi-view camera images, proprioceptive robot states, and actions. We include classic motor control and robotic tasks, namely, single object picking, bin picking, stacking, and destacking, with variations in object poses, shape, and appearance.

To understand the effect of pre-training, we perform a series of controlled real-world experiments comparing pre-training to training from scratch. We find that RPT consistently outperforms training from scratch and that the improvements are larger for harder tasks. We further find that our approach can benefit from pre-training data coming from different tasks, better vision encoders, and larger sensorimotor context lengths. Consistent with prior work on self-supervised pre-training in other domains [5], we find that fine-tuning is more effective than linear probe evaluation.

2 Related Work

Robot learning with self-supervision refers to the learning paradigm where robots gather data and learn directly from their interactions with their surrounding environments without human demonstrations or human-provided labels. Many prior works focus on learning policy for a single task. [7] learns visuomotor policy learning with self-supervised visual correspondence; [8, 9, 10, 11] learn general-purpose grasping policies based on grasp attempts; [12, 13] calibrate learned grasping policies for particular objects based on grasp success feedback. MT-Opt [14] simultaneously collects robot trajectories for multiple tasks and learns corresponding policies; [15] learns goal conditional robot policy from data collected by non-expert human demonstrators. In this work, instead of focusing on learning one or multiple task policies, we use self-supervision to learning sensorimotor representations that can help with learning downstream robot policies.

Supervised pre-training for robotics has been shown helpful in enabling generalization across robotics tasks. Shridhar et al. [16] adapt Contrastive Language–Image Pre-training (CLIP) [17] for learning language-conditioned pick-and-place policy. BC-Z [18], PerAct [19], and RT-1 [3] train language-conditioned behavior cloning policies from human or expert demonstrations that directly predict the next actions. Gato [20] performs autoregressive sequence modeling of multi-modal agent data. We share the goal of learning from sequences of multi-modal agent data but employ a masked objective, operate on latent visual representations, and focus more on an the physical world. PaLM-E [21] is an embodied Visual Question Answering (VQA) model and it uses a Vision Transformer [22] trained for novel-view synthesis to extract visual latent. The proposed method focuses on using a self-supervised objective to learn cross-modal representations of visual latents, proprioceptive states, and action information from unlabeled robot trajectories.

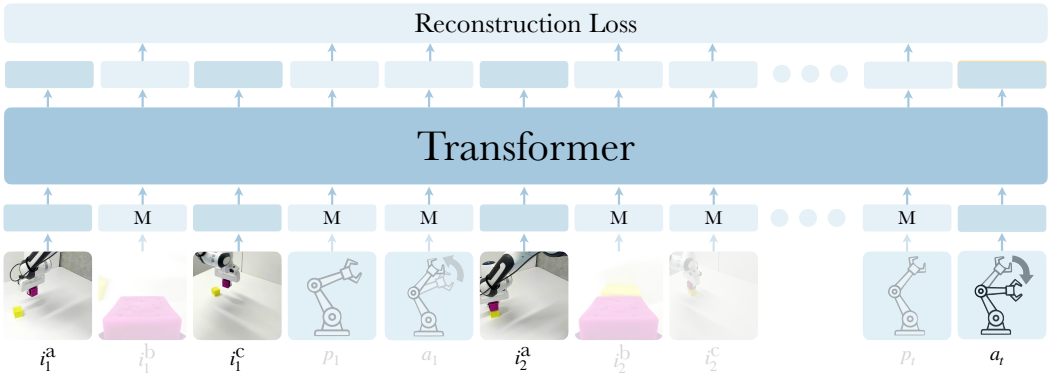


Figure 2: **Masked sensorimotor pre-training.** Given a sequence of camera images, proprioceptive robot states, and past robot actions, we randomly mask out different modalities at each time step in the sequence (opaque) and train a model to predict the masked content.

3 Robot Learning with Sensorimotor Pre-training

Our model, RPT, is a Transformer that operates on sensorimotor streams containing multi-vew camera images, proprioceptive robot states, and robot actions. We first pre-train the model on real-world robot trajectories with masked prediction (Figure 2). We then use the model for downstream robotic tasks in different evaluation modes corresponding to different mask prompts (Figure 3).

3.1 Masked Sensorimotor Pre-training

Vison latents. Our model must process sequences of high-dimension images from multiple views, which is challenging from both the learning and the computational perspective. To overcome this, we use a pre-trained vision model to compute visual representations. This enables us to leverage large quantities of visual data from diverse Internet videos that do not have proprioception and action. Moreover, it has important implications for real-world deployment. Namely, by encoding input images independently and operating on latent visual representations, we decouple the vision cost from the sensorimotor cost. This enables us to scale to large vision encoders, sensorimotor models, and context lengths, while maintaining fast training time and real-world inference speed.

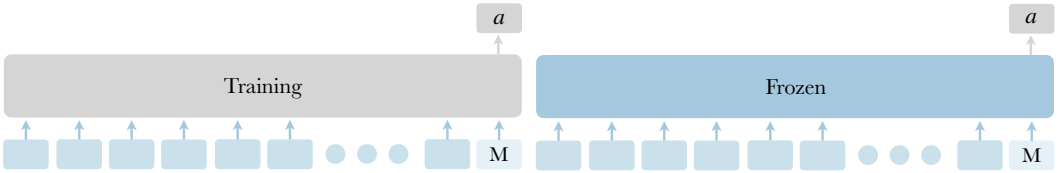
Token encoders. We use a separate linear encoder per modality. Since our modality encoders do not share weights, we do not use additional modality embeddings. To represent time, we add positional embeddings to each token. All tokens from a single timestep share the positional embedding value.

Token masking. Given a sequence of sensorimotor tokens we randomly mask out a subset. We represent each masked out input using a mask token that is learnt per modality and shared across time. We mask the input tokens independently without taking the time step or modality into consideration which encourages the model to learn to correlate information across both modalities and time.

Transformer model. The encoded and masked sequence of tokens is passed into a Transformer model [6]. We follow the standard Transformer design consisting of a series of Transformer blocks with multi-head self-attention operations. Our model predicts a set of fixed-length latent representations corresponding to each input token.

Prediction heads. We decode the hidden representations into prediction targets using linear project layers. Each latent is decoded from the hidden size back to the original modality dimension. We make predictions in the original continuous input space.

Masked objective. We compute the mean squared error reconstruction loss between the predictions and the ground truth input values. We apply the loss only to the predictions from the latent representations corresponding to the masked inputs. Predictions for the observed inputs incur no loss. To weight the importances and scale of different modalities we apply a per-modality loss weight.



(a) Finetune (b) Linear Probe

Figure 3: **Downstream evaluation.** We consider two different settings for evaluation on downstream tasks: (a) *Fine-Tune* fine-tunes the entire pre-trained model on the downstream task; (b) *Linear Probe* freezes the pre-trained model and trains a linear action read out layer.

3.2 Downstream Evaluation

The goal of pre-training is to learn general sensorimotor representations that can be used for solving downstream robotic tasks. Here, we focus on two evaluation modes that are most common in the self-supervised learning literature in language and computer vision [4, 5, 23, 24].

Fine-tuning. One way to use pre-trained representations is to fine-tune them with data from the downstream task (Figure 3a), which is particularly beneficial when the amount of downstream data is limited. In this setting pre-training can be seen as providing a good initialization for learning.

Linear probe. An alternate approach is to directly evaluate the pre-trained representations on downstream tasks (Figure 3b). We can achieve this by training only a linear action prediction layer on top of frozen sensorimotor representations, which we refer to as linear probe evaluation.

4 A Dataset of Sensorimotor Trajectories

To study our pre-training approach, we collect a large dataset of real-world robot trajectories. In this section we describe our data collection process and show qualitative examples from our dataset.

Hardware. We use the Franka robot with 7 degrees of freedom arm and a parallel jaw gripper (Figure 4, top left). We record proprioceptive information in the form of joint positions and velocities. We compute joint space actions as differences between consecutive states. The workspace is equipped with three high-quality Logitech Brio color cameras. One egocentric camera is attached to the robot hand, and two exocentric cameras are attached to the left and the right side of the robot. We synchronize the data streams from the cameras and the robot at 60 Hz and save data at 30 Hz.

(1) Pick: The robot is tasked to grasp and lift a cube (among $\mathcal{C}$ randomly colored cubes) that rests on the table in one of its 6 stable poses. We script a data generation scheme where the robot takes in the $\mathcal{C}$ cubes at randomly generated 3DoF poses in robot frame that are not in collision and generates sequences of 3DoF picks-and-places poses for each of the cubes consecutively. Then a scripted policy grasps the cube. After grasping, the robot places the grasped cube in the next randomly generated pose, moves back to the start joint configuration, and starts the next grasp.

(2) Bin Pick: The robot picks items from a gray bin filled with randomly-placed, both soft and rigid objects (see Fig. 4). We use an overhead L515 Realsense Depth Camera to capture depth images of the workspace and feed the depth image into a Fully Convolutional-Grasp Quality Convolutional Neural Network (FC-GQCNN) [25] to generate grasps. Each grasp starts with the robot moving to a joint configuration that does not occlude the depth camera from viewing the workspace. After FC-GQCNN generates a grasp pose, the scripted policy described in (a) is used to pick the object. If the gripper is not fully closed the end of the trajectory, we consider the trajectory successful.

(3) Stack and (4) Destack: Starting with two randomly colored cubes resting on the table, the robot is tasked to first move one of the cubes above the other cube, and then move the stacked cube back onto the table. The cube poses are generated similarly to (1). For stacking, the height of the place pose is offset by the height of the cube; similarly, for destacking, the pick pose height is offset by the cube's height. A scripted policy then executes the pick-and-place trajectory.

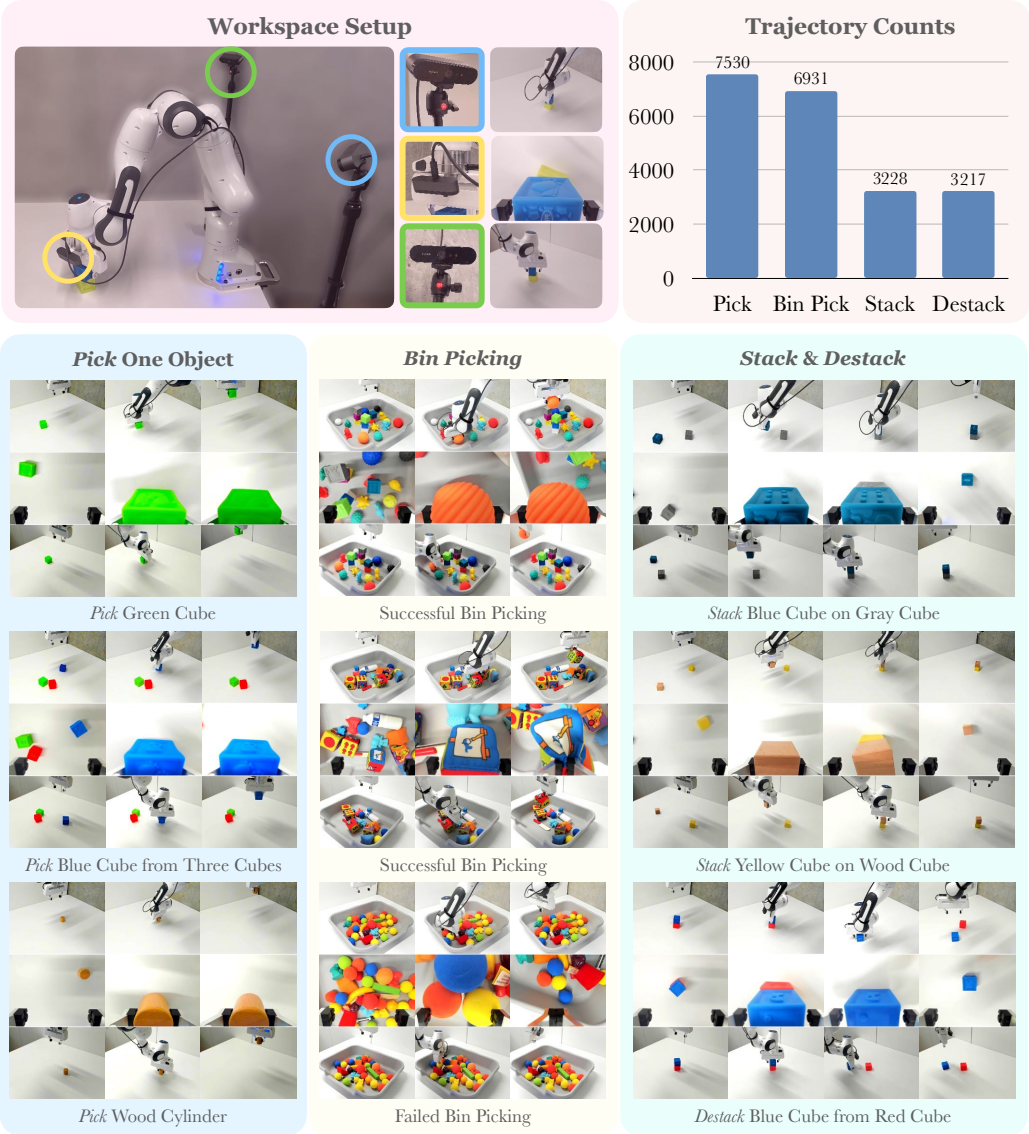


Figure 4: **Dataset.** We collect a dataset of over 20,000 real-world trajectories using a Franka robotic arm. Each trajectory is a sequence of high-quality images from three cameras (one attached to the hand and two on the sides), proprioceptive robot states, and actions. We consider picking, bin bicking, destacking, and stacking tasks, with variations in object position, shape, and appearance.

Statistics. We collect ~20,000 real-world trajectories over the course of 9 months (Figure 4). The dataset includes ~7,000 trajectories for both single object and bin picking, as well as ~3,000 trajectories for stacking and destacking each. The average length of picking and destacking trajectories is ~300 while the stacking trajectories are longer at ~600 steps. The dataset contains variations in object poses, shape, and appearance. We show frames from example trajectories in Figure 4.

5 Implementation Details

Vision encoder. We use the standard Vision Transformer (ViT) architecture [26] to encode image inputs and evaluate RPT using two versions of ViT: ViT-Base and ViT-Large, with 86M and 307M parameters respectively. We use the pre-trained models from [1] which were trained via MAE [5] on a collection of 4.5M images from Ego4D [27], Epic [28], Something-Something [29], 100 Days of Hands [30], and ImageNet [31]. We extract features from all three cameras using the same model.

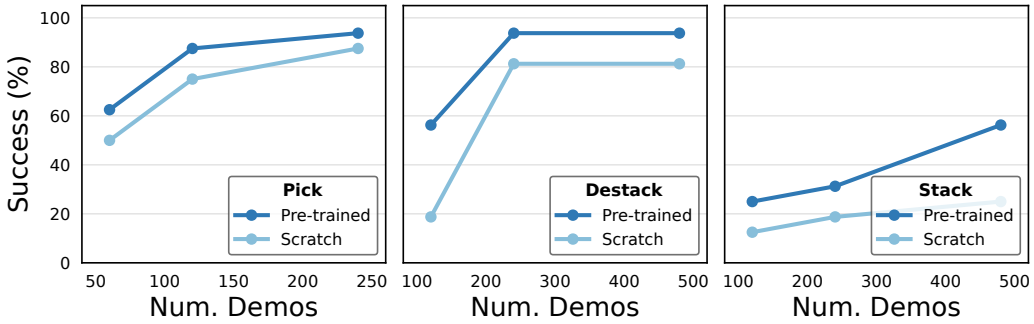


Figure 5: **Sample complexity, fine-tuning.** We study the effect of pre-training as the number of fine-tuning demonstrations increases. We observe that pre-training brings consistent improvements over training from scratch and that the gains are larger for the hardest stacking task.

Sensorimotor transformer. For the sensorimotor model, we use an encoder-only Transformer with about 1M parameters. The transformer has a hidden dimension of 192 and contains four transformer blocks, each with 4 heads and mlp ratio of 2.0. The transformer can take sensorimotor input with varied context length. We try input with 1, 4, and 16 steps and get the best empirical performances out of context length of 16 (see Section 6.3).

Pre-training. We adopt a uniform random masking strategy, *i.e.*, for each token, we first randomly sample a masking probability from a preset range (*e.g.*, $[0.7, 0.9]$), and then mask the token with the corresponding probability. We consider three types of masking: (i) **token masking** independently masks each token across different time steps and modalities, (ii) **time-step masking** masks the tokens from all the modalities within the same time step, and (iii) **modality masking** masks the tokens from all time steps within the same modality. See Figure 8 for an illustration. We conduct ablation studies on different masking probabilities and masking types in Section 6.3.

Fine-tuning. We fine-tune the pre-trained model by imitation learning on the downstream tasks. During fine-tuning, the model takes the past sensorimotor inputs with the same context length as in pre-training, with the action token in the last time step replaced with a mask token. Then we train a linear head that predicts the next 16 steps of actions from the last output mask token.

6 Experiments

In this section, we conduct empirical evaluations to show **(i)** the effectiveness of pre-training (Section 6.1), **(ii)** the generalizability of RPT to various pre-training data (Section 6.2), and **(iii)** ablations of core design decisions (Section 6.3). For each evaluation result, we report task success rate across 16 real-world trials with variations in object positions and rotations.

6.1 Comparisons to Training from Scratch

We first show the effectiveness of our masked pre-training by comparing fine-tuning performances with or without pre-training. We keep the same vision encoders, sensorimotor architectures, and optimize the learning rate, batch size, and training length per model. We consider three evaluation tasks of increasing difficulty: picking, destacking, and stacking. For pre-training, we use different variants of data from the same task as in fine-tuning and hold out an unseen set for evaluation.

In Figure 5 we show the performance as the number of fine-tuning demonstrations increases. We observe that pre-training leads to consistent improvements over training from scratch, across tasks and data regimes. Moreover, the gains are largest for the hardest stacking task. This is a long horizon task that requires precise spatial localization and fine motor control which is particularly challenging for joint position control approaches like we use here.

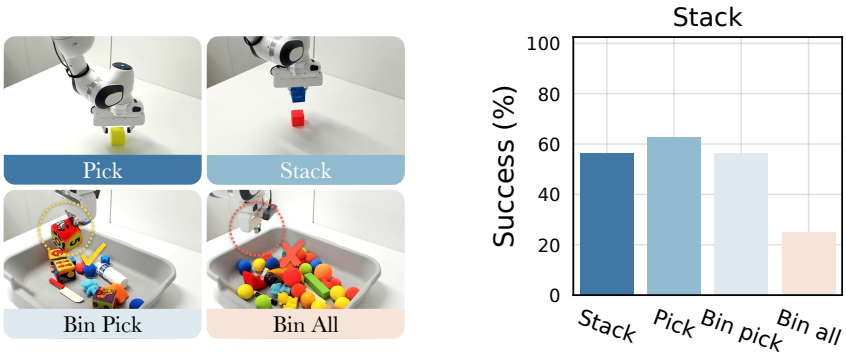


Figure 6: **Pre-training data.** We evaluate the performance of models pre-trained on different data distributions (left) on the stacking task (right). We observe that pre-training with successful trajectories leads to consistent improvements while including unsuccessful trajectories is not as effective.

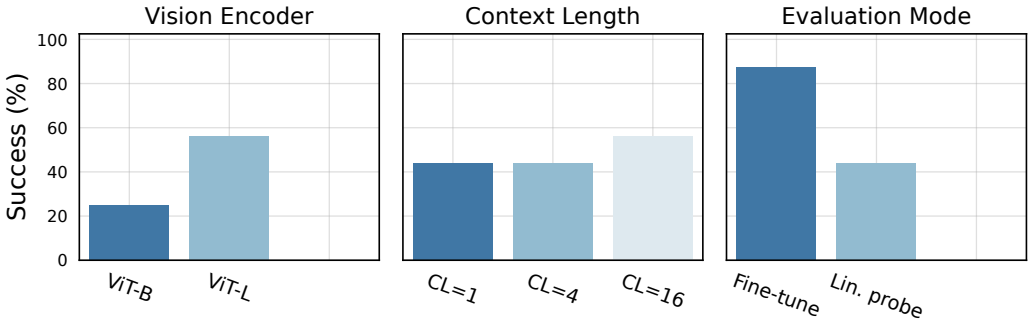


Figure 7: **Ablations.** We find that our approach benefits from better vision encoders (left), larger temporal context lengths (middle), and that fine-tuning is more effective than linear probe (right).

6.2 Generalizability to Different Pre-Training Data

Previously, we use data from the same task for pre-training and fine-tuning. To verify the masked pre-training actually learns a general sensorimotor representation instead of representation for the specific task, we test the performance when pre-training on a different task from fine-tuning. Specifically, we first pre-train our model on data from different tasks: stacking, picking, successful bin picking trajectories, and all bin picking trajectories (Figure 6, left). We then fine-tune and evaluate the model on stacking. In Figure 6, right, we see that RPT obtains similar performance on stacking when pre-trained on stacking, picking, or bin picking, indicating that our approach is able to effectively learn sensorimotor representations from data of different tasks. We also observe a lower performance when pre-trained on imperfect (both successful and failed) trajectories of bin picking, indicating the importance of pre-training data quality.

6.3 Ablation Studies

Vision encoder. We study the importance of visual feature by comparing vision encoders with different sizes. Specifically, we test ViT-B and ViT-L on the task of stacking. Results are shown in Figure 7, left. We observe a significant improvement of success rate when switching ViT-B to ViT-L, which is probably because stacking requires precise spatial localization and the features of ViT-L contain more spatial information. It is worth noting that our model is still capable of real-time inference even with ViT-L as the visual encoder which contains over 300M parameters.

Context length. We compare sensorimotor models with different input context length. We test context length of 1, 4, and 16 on the task of stack. As shown in Figure 7, middle, RPT obtains similar performance with 1-step and 4-step input, while having higher success rate with context length of 16. This highlights the scalability of RPT on longer context length.

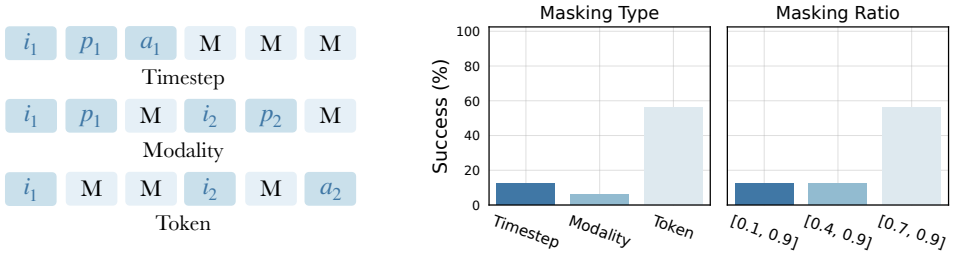


Figure 8: **Masking ablations.** We find that masking across all tokens works considerably better than masking all tokens from a timestep or all tokens from a modality at the time. Moreover, we find that a high masking ratio is essential for learning good sensorimotor representations (right).

Linear probe. To evaluate the representation learned from masked pre-training, we freeze the pre-trained representation and test the linear probing performance of RPT. We train a linear head on top of the representation using 120 demos of picking. As shown in Figure 7, right, linear probing reaches a success rate of 43.75% which is comparable to the baseline trained on 60 demos from scratch (Figure 5). This indicates the representation learned from masked pre-training already contains rich information for motion planning.

Masking type. At the core of our pre-training approach is the masked sensorimotor prediction task. We hypothesize that the masked pre-training facilitates downstream tasks by learning structured correlation across both time and modality. To verify this, we conduct ablation on only learning the modality or temporal correlation. To be specific, we test the time-step masking which masks the tokens of different modalities within one time step, and modality masking which masks the tokens from all time steps within one modality. These remove the modality and temporal correlation, respectively. Results are shown in Figure 8. We can see that both time-step and modality masking obtains a poor performance compared to token masking which independently masks each token across time and modality. This provides a supportive evidence to our hypothesis that both temporal and modality correlation are crucial for learning downstream tasks.

Masking ratio. We evaluate different choices of masking ratio. Previous work on masked visual pre-training [5] finds a high masking ratio is helpful. In our experiments we also have a similar observation. As shown in Figure 8, there is a significant performance drop when masking ratio is between $[0.1, 0.9]$ or $[0.4, 0.9]$ compared to $[0.7, 0.9]$. This indicates a high redundancy between different time steps and modalities in the input sensorimotor sequences and a high masking ratio is crucial for learning useful representations for downstream tasks.

Inference speed. Our sensorimotor model is designed to operate on latent visual representations that are computed per image, which decouples the vision model from the sensorimotor model making the vision computation cost independent of the sensorimotor context length. This enables us to use large vision models (up to 307M parameters) while running inference at 10 Hz. To put in context, an alternate robotic Transformer design, RT-1 [3], uses a 14M parameter model and runs at 3 Hz.

7 Discussion

Limitations. To study our pre-training approach we collect a dataset of physical sensorimotor trajectories on a single robot in a single lab. While this is a good initial step that helps us understand our method with controlled experiments, it has important limitations in terms of diversity and realism. Scaling our method to more diverse environments remains an important area for future research.

Conclusion. We describe an approach for robot learning with sensorimotor pre-training. Our model is a Transformer that operates on sequences of sensorimotor tokens. We pre-train our model by masked prediction. When evaluated on downstream tasks, we find that pre-training leads to consistent improvements over training from scratch. Finally, we show favorable scaling properties.

References

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

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

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

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HCT*, 2019.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[7] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *RA-L*, 2019.

[8] 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.

[9] 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. *IJRR*, 2018.

[10] X. Zhu, D. Wang, O. Biza, G. Su, R. Walters, and R. Platt. Sample efficient grasp learning using equivariant models. *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[11] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, 2016.

[12] M. Danielczuk, A. Balakrishna, D. S. Brown, S. Devgon, and K. Goldberg. Exploratory grasping: Asymptotically optimal algorithms for grasping challenging polyhedral objects. *arXiv preprint arXiv:2011.05632*, 2020.

[13] L. Fu, M. Danielczuk, A. Balakrishna, D. S. Brown, J. Ichnowski, E. Solowjow, and K. Goldberg. Legs: Learning efficient grasp sets for exploratory grasping. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8259–8265. IEEE, 2022.

[14] D. Kalashnkov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv*, 2021.

[15] Z. J. Cui, Y. Wang, N. Muhammad, L. Pinto, et al. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv:2210.10047*, 2022.

[16] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2021.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[18] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, 2022.

[19] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[20] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv:2205.06175*, 2022.

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

[22] M. S. M. Sajjadi, D. Duckworth, A. Mahendran, S. van Steenkiste, F. Pavetić, M. Lučić, L. J. Guibas, K. Greff, and T. Kipf. Object Scene Representation Transformer. *NeurIPS*, 2022.

[23] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[25] V. Satish, J. Mahler, and K. Goldberg. On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks. *IEEE Robotics and Automation Letters*, 2019.

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

[27] 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. *arXiv:2110.07058*, 2021.

[28] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 2021.

[29] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

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

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.