Masked World Models for Visual Control

Anonymous Author(s)
Affiliation
Address
email

Abstract: We introduce a visual model-based reinforcement learning (RL) framework that decouples visual representation learning and dynamics learning. For visual representation learning, we train an autoencoder with convolutional layers and vision transformers (ViT) to reconstruct pixels given masked convolutional features, inspired by the recent success of self-supervised learning with ViT and masked image modeling. Moreover, in order to encode task-relevant information, we introduce an auxiliary reward prediction objective for the autoencoder. For dynamics learning, we learn a latent dynamics model that operates on visual representations from the autoencoder. Our framework continually updates both the autoencoder and dynamics model using online samples collected from environment interaction. We demonstrate that our decoupling approach achieves state-of-the-art performance on a variety of visual robotic tasks from Meta-world and RLBench, *e.g.*, we achieve 81.7% success rate on 50 visual robotic manipulation tasks from Meta-world, while the baseline achieves 67.9%.

Keywords: Visual model-based RL, World models

1 Introduction

Model-based reinforcement learning (RL) holds the promise of sample-efficient robot learning by learning a world model and leveraging it for planning [1, 2, 3] or generating imaginary states for behavior learning [4, 5]. These approaches have also previously been applied to environments with visual observations, by learning an action-conditional video prediction model [6, 7] or a latent dynamics model that predicts compact representations in an abstract latent space [8, 9]. However, learning world models on environments with complex visual observations, *e.g.,* modeling interactions between multiple small objects, as accurately as state-based world models still remains a challenge.

We argue that this difficulty comes from the design of current approaches that typically learn a single model to excel at learning both visual representations and dynamics [9, 10]. This imposes a trade-off between visual representation learning and dynamics learning, thus making it difficult to learn world models in environments where visual representation learning alone is already a challenging problem. In contrast, Ha and Schmidhuber [11] investigated the approach that separately trains a variational autoencoder (VAE) [12] and a dynamics model on top of the VAE features. By enabling us to leverage any visual representation learning and dynamics learning scheme, this approach has the potential to be a generic approach for learning world models. However, it is also limited in that VAE representations may not be amenable to dynamics learning [8, 10] or may not capture task-relevant information [11].

We present MWM: **M**asked **W**orld **M**odels, a visual model-based RL framework that decouples visual representation learning and dynamics learning by learning a latent dynamics model on top of a self-supervised vision transformer (ViT) [13], inspired by the recent success of self-supervised learning methods with ViT and masked image modeling [14, 15]. Specifically, we separately update visual representations and dynamics by repeating the iterative processes of (i) training an autoencoder with convolutional feature masking and reward prediction, and (ii) learning a latent dynamics model that predicts visual representations from the autoencoder (see Figure 1).

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

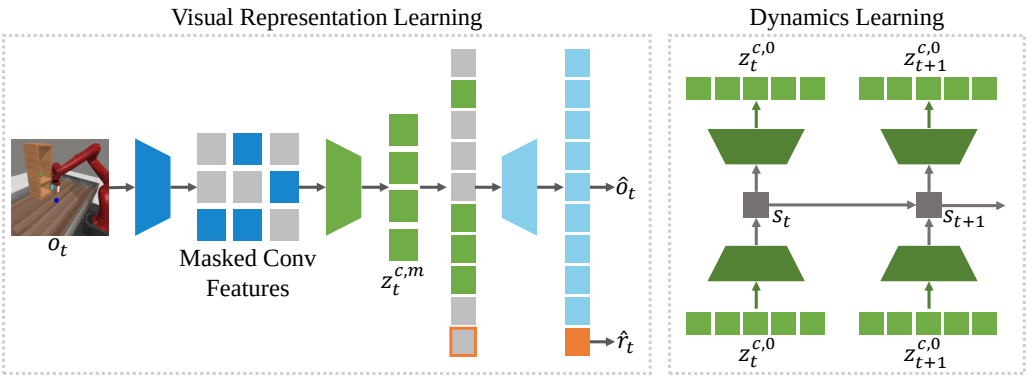


Figure 1: Illustration of our approach. We continually update visual representations and dynamics using online samples collected from environment interaction, by repeating iterative processes of training (Left) an autoencoder with convolutional feature masking and reward prediction and (Right) a latent dynamics model in the latent space of the autoencoder. We note that autoencoder parameters are not updated during dynamics learning.

Contributions We highlight the contributions of our paper below:

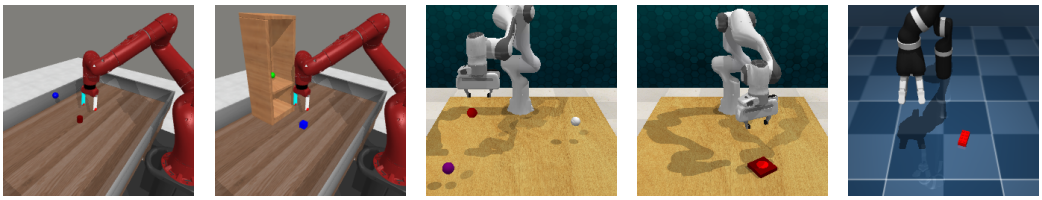
- We demonstrate the effectiveness of decoupling visual representation learning and dynamics learning for visual model-based RL. MWM significantly outperforms a state-of-the-art model-based baseline [16] on various visual control tasks from Meta-world [17] and RLBench [18].
- We show that a self-supervised ViT trained to reconstruct visual observations with convolutional feature masking can be effective for visual model-based RL. Interestingly, we find that masking convolutional features can be more effective than pixel patch masking [14], by allowing for capturing fine-grained details within patches. This is in contrast to the observation in Touvron et al. [19], where both perform similarly on the ImageNet classification task [20].
- We show that an auxiliary reward prediction task can significantly improve performance by encoding task-relevant information into visual representations.

2 Related Work

World models from visual observations There have been several approaches to learn visual representations for model-based via image reconstruction [6, 7, 8, 9, 10, 11, 16, 21, 22], *e.g.,* learning a video prediction model [6] or a latent dynamics model [8, 9, 10]. This has been followed by a series of works that demonstrated the effectiveness of model-based approaches for solving video games [16, 23, 24] and visual robot control tasks [7, 22, 25, 26]. There also have been several works that considered different objectives, including bisimulation [27] and contrastive learning [28, 29, 30]. While most prior works introduce a single model that learns both visual representations and dynamics, we instead develop a framework that decouples visual representation learning and dynamics learning.

Self-supervised vision transformers Self-supervised learning with vision transformers (ViT) [13] has been actively studied. For instance, Chen et al. [31] introduced MoCo-v3 which trains a ViT with contrastive learning. Caron et al. [32] introduced DINO which utilizes a self-distillation loss [33], and demonstrated that self-supervised ViTs contain information about the semantic layout of images. Training self-supervised ViTs with masked image modeling [14, 15, 34, 35, 36, 37, 38] has also been successful. In particular, He et al. [14] proposed a masked autoencoder (MAE) that reconstructs masked pixel patches with an asymmetric encoder-decoder architecture. Unlike MAE, we propose to randomly mask features from early convolutional layers [39] instead of pixel patches and demonstrate that self-supervised ViTs can also be effective for visual model-based RL.

We provide more discussion on related works in more detail in Appendix C.



(a) Pick Place (b) Shelf Place (c) Reach Target (d) Push Button (e) Reach Duplo

Figure 2: Examples of visual observations used in our experiments. We consider a variety of visual robot control tasks from Meta-world [17], RLBench [18], and DeepMind Control Suite [40].

3 Preliminaries

Problem formulation We formulate a visual control task as a partially observable Markov decision process (POMDP) [41], which is defined as a tuple $(\mathcal{O}, \mathcal{A}, p, r, \gamma)$. $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $p\left(o_t | o_{<t}, a_{<t}\right)$ is the transition dynamics, r is the reward function that maps previous observations and actions to a reward $r_t = r\left(o_{\leq t}, a_{<t}\right)$, and $\gamma \in [0, 1)$ is the discount factor.

Dreamer Dreamer [16, 22] is a visual model-based RL method that learns world models from pixels and trains an actor-critic model via latent imagination. Specifically, Dreamer learns a Recurrent State Space Model (RSSM) [9], which consists of following four components:

$$\begin{aligned} &\text{Representation model:} \quad s_t \sim q_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t) \quad &&\text{Image decoder:} \quad \hat{o}_t \sim p_\theta(\hat{o}_t \mid s_t) \\ &\text{Transition model:} \quad \hat{s}_t \sim p_\theta(\hat{s}_t \mid s_{t-1}, a_{t-1}) \quad &&\text{Reward predictor:} \quad \hat{r}_t \sim p_\theta(\hat{r}_t \mid s_t) \end{aligned} \tag{1}$$

The representation model extracts model state s_t from previous model state s_{t-1}, previous action a_{t-1}, and current observation o_t. The transition model predicts future state $\hat{s}_t$ without the access to current observation o_t. The image decoder reconstructs raw pixels to provide learning signal, and the reward predictor enables us to compute rewards from future model states without decoding future frames. All model parameters θ are trained to jointly learn visual representations and environment dynamics by minimizing the negative variational lower bound [12]:

$$\begin{aligned} &\mathcal{L}(\theta) \doteq \mathbb{E}_{q_\theta(s_{1:T} | a_{1:T}, o_{1:T})} \Big[\\ &\textstyle\sum_{t=1}^{T} \Big(-\ln p_\theta(o_t | s_t) - \ln p_\theta(r_t | s_t) + \beta \, \mathrm{KL}\left[q_\theta(s_t | s_{t-1}, a_{t-1}, o_t) \,\|\, p_\theta(\hat{s}_t | s_{t-1}, a_{t-1}) \right] \Big) \Big], \end{aligned} \tag{2}$$

where β is a hyperparameter that controls the tradeoff between the quality of visual representation learning and the accuracy of dynamics learning [42]. Then, the critic is learned to regress the values computed from imaginary rollouts, and the actor is trained to maximize the values by propagating analytic gradients back through the transition model (see Appendix A for the details).

Masked autoencoder Masked autoencoder (MAE) [14] is a self-supervised visual representation technique that trains an autoencoder to reconstruct raw pixels with randomly masked patches consisting of pixels. Following a scheme introduced in vision transformer (ViT) [13], the observation $o_t \in \mathbb{R}^{H \times W \times C}$ is processed with a patchify stem that reshapes o_t into a sequence of 2D patches $h_t \in \mathbb{R}^{N \times (P^2 C)}$, where P is the patch size and $N = HW/P^2$ is the number of patches. Then a subset of patches is randomly masked with a ratio of m to construct $h_t^m \in \mathbb{R}^{M \times (P^2 C)}$.

$$\text{Patchify stem:} \quad h_t = f_\phi^{\texttt{patch}}(o_t) \quad \text{Masking:} \quad h_t^m \sim p^{\texttt{mask}}(h_t^m \mid h_t, m) \tag{3}$$

A ViT encoder embeds only the remaining patches h_t^m into D-dimensional vectors, concatenates the embedded tokens with a learnable CLS token, and processes them through a series of Transformer layers [43]. Finally, a ViT decoder reconstructs the observation by processing tokens from the encoder and learnable mask tokens through Transformer layers followed by a linear output head:

$$\text{ViT encoder:} \quad z_t^m \sim p_\phi(z_t^m \mid h_t^m) \quad \text{ViT decoder:} \quad \hat{o}_t \sim p_\phi(\hat{o}_t \mid z_t^m) \tag{4}$$

All the components paramaterized by ϕ are jointly optimized to minimize the mean squared error (MSE) between the reconstructed and original pixel patches. MAE computes z_t^0 without masking, and utilizes its first component (*i.e.,* CLS representation) for downstream tasks (*e.g.,* image classification).

4 Method

In this section, we present Masked World Models (MWM), a visual model-based RL framework for learning accurate world models by separately learning visual representations and environment dynamics. Our method repeats (i) updating an autoencoder with convolutional feature masking and an auxiliary reward prediction task (see Section 4.1), (ii) learning a dynamics model in the latent space of the autoencoder (see Section 4.2), and (iii) collecting samples from environment interaction. We provide the overview and pseudocode of MWM in Figure 1 and Appendix D, respectively.

4.1 Autoencoders with Convolutional Feature Masking and Reward Prediction

It has been observed that masked image modeling with a ViT architecture [14, 15, 35] enables compute-efficient and stable self-supervised visual representation learning. This motivates us to adopt this approach for visual model-based RL, but we find that masked image modeling with commonly used pixel patch masking [14] often makes it difficult to learn fine-grained details within patches, *e.g.,* small objects (see Appendix B for a motivating example). While one can consider small-size patches, this would increase computational costs due to the quadratic complexity of self-attention layers.

To handle this issue, we instead propose to train an autoencoder that reconstructs raw pixels given randomly masked convolutional features. Unlike previous approaches that utilize a patchify stem and randomly mask pixel patches (see Section 3), we adopt a convolution stem [13, 39] that processes o_t through a series of convolutional layers followed by a flatten layer, to obtain $h_t^c \in \mathbb{R}^{N_c \times D}$ where N_c is the number of convolutional features. Then h_t^c is randomly masked with a ratio of m to obtain $h_t^{c,m} \in \mathbb{R}^{M_c \times D}$, and ViT encoder and decoder process $h_t^{c,m}$ to reconstruct raw pixels.

$$\begin{aligned} &\text{Convolution stem:} \quad h_t^c = f_\phi^{\texttt{conv}}(o_t) && \text{Masking:} \quad h_t^{c,m} \sim p^{\texttt{mask}}(h_t^{c,m} \mid h_t^c, m) \\ &\text{ViT encoder:} \quad z_t^{c,m} \sim p_\phi(z_t^{c,m} \mid h_t^{c,m}) && \text{ViT decoder:} \quad \hat{o}_t \sim p_\phi(\hat{o}_t \mid z_t^{c,m}) \end{aligned} \tag{5}$$

Because early convolutional layers mix low-level details, we find that our autoencoder can effectively reconstruct all the details within patches by learning to extract information from nearby non-masked features (see Figure 7 for examples). This enables us to learn visual representations capturing such details while also achieving the benefits of MAE, *e.g.,* stability and compute-efficiency.

Reward prediction In order to encode task-relevant information that might not be captured solely by the reconstruction objective, we introduce an auxiliary objective for the autoencoder to predict rewards jointly with pixels. Specifically, we make the autoencoder predict the reward r_t from $z_t^{c,m}$ in conjunction with raw pixels.

$$\text{ViT decoder with reward prediction:} \quad \hat{o}_t, \hat{r}_t \sim p_\phi(\hat{o}_t, \hat{r}_t \mid z_t^{c,m}) \tag{6}$$

In practice, we concatenate one additional learnable mask token to inputs of the ViT decoder, and utilize the corresponding output representation for predicting the reward with a linear output head.

High masking ratio Introducing early convolutional layers might impede the masked reconstruction tasks because they propagate information across patches [19], and the model can exploit this to find a shortcut to solve reconstruction tasks. However, we find that a high masking ratio (*i.e.,* 75%) can prevent the model from finding such shortcuts and induce useful representations (see Figure 6(b) for supporting experimental results). This also aligns with the observation from Touvron et al. [19], where masked image modeling [15] with a convolution stem [44] can achieve competitive performance with the patchify stem on the ImageNet classification task [20].

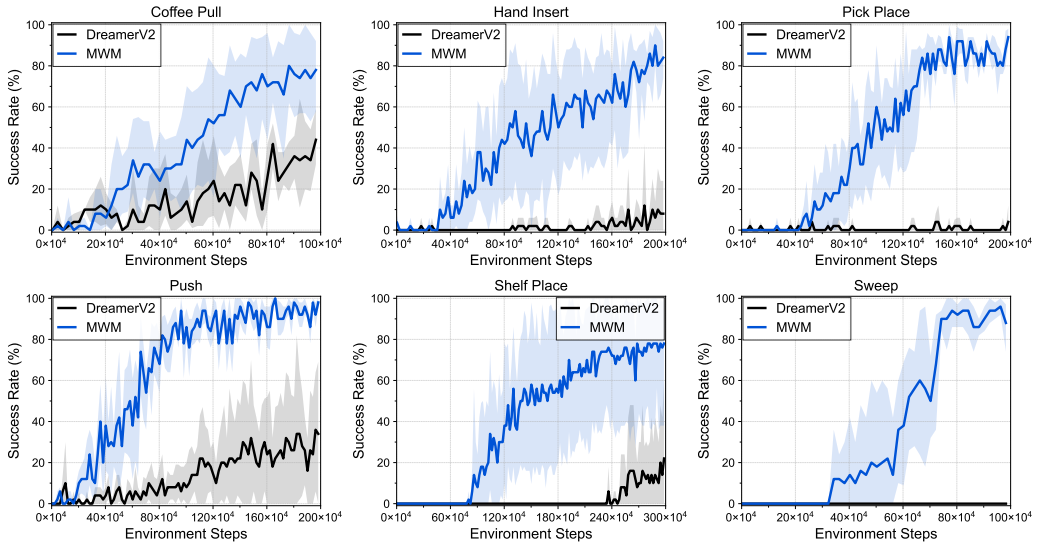


Figure 3: Learning curves on six visual robotic manipulation tasks from Meta-world as measured on the success rate. We select the tasks that require modeling interactions between small objects and robot arms. Learning curves on 50 tasks are available in Appendix G. The solid line and shaded regions represent the mean and bootstrap confidence intervals, respectively, across five runs.

4.2 World Models in Latent Space

Once we learn visual representations, we leverage them for efficiently learning a dynamics model in the latent space of the autoencoder. Specifically, we obtain the frozen representations $z_t^{c,0}$ from the autoencoder, and then train a variant of RSSM whose inputs and reconstruction targets are $z_{t,0}^{c}$, by replacing the representation model and the image decoder in Equation 1 with following components:

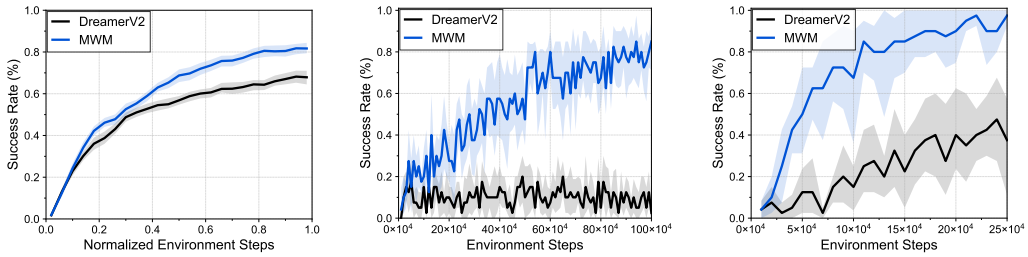
$$\begin{aligned} &\text{Representation model:} && s_t \sim q_\theta(s_t \mid s_{t-1}, a_{t-1}, z_t^{c,0}) \\ &\text{Visual representation decoder:} && \hat{z}_t^{c,0} \sim p_\theta(\hat{z}_t^{c,0} \mid s_t) \end{aligned} \tag{7}$$

Because visual representations capture both high- and low-level information in an abstract form, the model can focus more on dynamics learning by reconstructing them instead of raw pixels (see Section 5.5 for relevant discussion). Here, we also note that we utilize all the elements of $z_t^{c,0}$ unlike MAE that only utilizes CLS representation for downstream tasks. We empirically find this enables the model to receive rich learning signals from reconstructing all the representations containing spatial information (see Appendix I for supporting experiments).

5 Experiments

We evaluate MWM on various robotics benchmarks, including Meta-world [17] (see Section 5.1), RLBench [18] (see Section 5.2), and DeepMind Control Suite [45] (see Section 5.3). We remark that these benchmarks consist of diverse and challenging visual robotic tasks. We also analyze algorithmic design choices in-depth (see Section 5.4) and provide a qualitative analysis of how our decoupling approach works by visualizing the predictions from the latent dynamics model (see Section 5.5).

Implementation We use visual observations of $64 \times 64 \times 3$. For the convolution stem, we stack 3 convolution layers with the kernel size of 4 and stride 2, followed by a linear projection layer. We use a 4-layer ViT encoder and a 3-layer ViT decoder. We find that initializing the autoencoder with a warm-up schedule at the beginning of training is helpful. Unlike MAE, we compute the loss on entire pixels because we do not apply masking to pixels. For world models, we build our implementation on top of DreamerV2 [16]. To take a sequence of autoencoder representations as inputs, we replace a CNN encoder and decoder with a 2-layer Transformer encoder and decoder. We use same hyperparameters within the same benchmark. More details are available in Appendix E.



(a) Meta-world aggregated (b) RLBench: Reach Target (c) RLBench: Push Button

Figure 4: (a) Aggregate performance on all 50 Meta-world tasks. We normalize environment steps by maximum steps in each task. The solid line and shaded regions represent the mean and stratified bootstrap confidence intervals, respectively, across 250 runs. We report the learning curves on (b) Reach Target and (c) Push Button from RLBench. Performances are not directly comparable to previous results [46, 47] due to the difference in setups (see Section 5.2). The solid line and shaded regions represent the mean and bootstrap confidence intervals, respectively, across eight runs.

5.1 Meta-world Experiments

Environment details In order to use a single camera viewpoint consistently over all 50 tasks, we use the modified `corner2` camera viewpoint for all tasks. In our experiments, we classify 50 tasks into `easy`, `medium`, `hard`, and `very hard` tasks where experiments are run over 500K, 1M, 2M, 3M environments steps with action repeat of 2, respectively. More details are available in Appendix F.

Results In Figure 3, we report the performance on a set of selected six challenging tasks that require agents to control robot arms to interact with small objects. We find that MWM significantly outperforms DreamerV2 in terms of both sample-efficiency and final performance. In particular, MWM achieves $> 80\%$ success rate on Pick Place while DreamerV2 struggles to solve the task. These results show that our approach of separating visual representation learning and dynamics learning can learn accurate world models on challenging domains. Figure 4(a) shows the aggregate performance over all the 50 tasks from the benchmark, demonstrating that our method consistently outperforms DreamerV2 overall. We also provide learning curves on all individual tasks in Appendix G, where MWM consistently achieves similar or better performance on most tasks.

5.2 RLBench Experiments

Environment details In order to evaluate our method on more challenging visual robotic manipulation tasks, we consider RLBench [18], which has previously acted as an effective proxy for real-robot performance [47]. Since RLBench consists of sparse-reward and challenging tasks, solving them typically requires expert demonstrations, specialized network architectures, additional inputs (e.g., point cloud and proprioceptive states), and an action mode that requires path planning [46, 47, 48, 49]. While we could utilize some of these components, we instead leave this as future work in order to maintain a consistent evaluation setup across multiple domains. In our experiments, we instead consider two relatively easy tasks with dense rewards, and utilize an action mode that specifies the delta of joint positions. We provide more details in Appendix F.

Results As shown in Figure 4(b) and Figure 4(c), we observe that our approach can also be effective on RLBench tasks, significantly outperforming DreamerV2. In particular, DreamerV2 achieves $< 20\%$ success rate on Reach Target, while our approach can solve the tasks with $> 80\%$ success rates. We find that this is because DreamerV2 fails to capture target positions in visual observations, while our method can capture such details (see Section 5.5 for relevant discussion and visualizations). However, we also note that these results are preliminary because they are still too sample-inefficient to be used for real-world scenarios. We provide more discussion in Section 6.

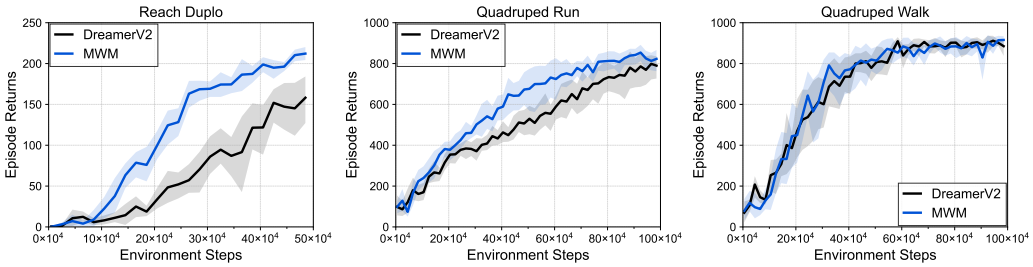
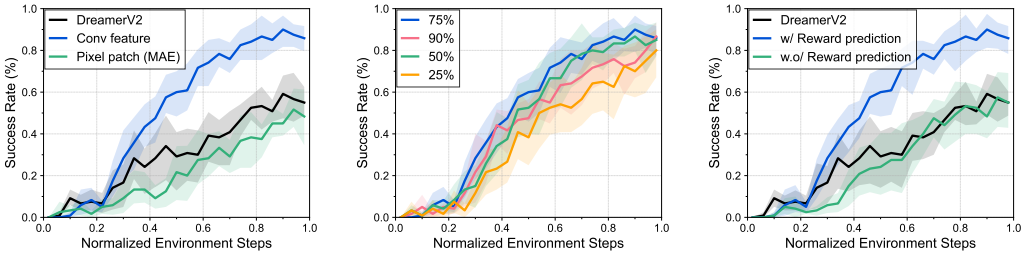


Figure 5: Learning curves on three visual robot control tasks from DeepMind Control Suite as measured on the episode return. The solid line and shaded regions represent the mean and bootstrap confidence intervals, respectively, across eight runs.



(a) Feature masking (b) Masking ratio (c) Reward prediction

Figure 6: Learning curves on three manipulation tasks from Meta-world that investigate the effect of (a) convolutional feature masking, (b) masking ratio, and (c) reward prediction. The solid line and shaded regions represent the mean and stratified bootstrap confidence interval across 12 runs.

5.3 DeepMind Control Suite Experiments

Environment details In order to demonstrate that our approach is generally applicable to diverse visual control tasks, we also evaluate our method on visual locomotion tasks from the widely used DeepMind Control Suite benchmark. Following a standard setup in Hafner et al. [22], we use an action repeat of 2 and default camera configurations. We provide more details in Appendix F.

Results Figure 5 shows that our method achieves competitive performance to DreamerV2 on visual locomotion tasks (i.e., Quadruped tasks), demonstrating the generality of our approach across diverse visual control tasks. We also observe that our method outperforms DreamerV2 on Reach Duplo, which is one of a few manipulation tasks in the benchmark (see Figure 2(e) for an example). This implies that our method is effective on environments where the model should capture fine-grained details like object positions. More results are available in Appendix H, where trends are similar.

5.4 Ablation Study

Convolutional feature masking We compare convolutional feature masking with pixel masking (*i.e.,* MAE) in Figure 6(a), which shows that convolutional feature masking significantly outperforms pixel masking. This demonstrates that enabling the model to capture fine-grained details within patches can be important for visual control. We also report the performance with varying masking ratio $m \in \{0.25, 0.5, 0.75, 0.9\}$ in Figure 6(b). As we discussed in Section 4.1, we find that $m = 0.75$ achieves better performance than $m \in \{0.25, 0.5\}$ because strong regularization can prevent the model from finding a shortcut from input pixels. However, we also find that too strong regularization (*i.e.,* $m = 0.9$) degrades the performance.

Reward prediction In Figure 6(c), we find that performance significantly degrades without reward prediction, which shows that the reconstruction objective might not be sufficient for learning task-relevant information. It would be an interesting future direction to develop a representation learning scheme that learns task-relevant information without rewards because they might not be available in practice. We provide more ablation studies and learning curves on individual tasks in Appendix I.

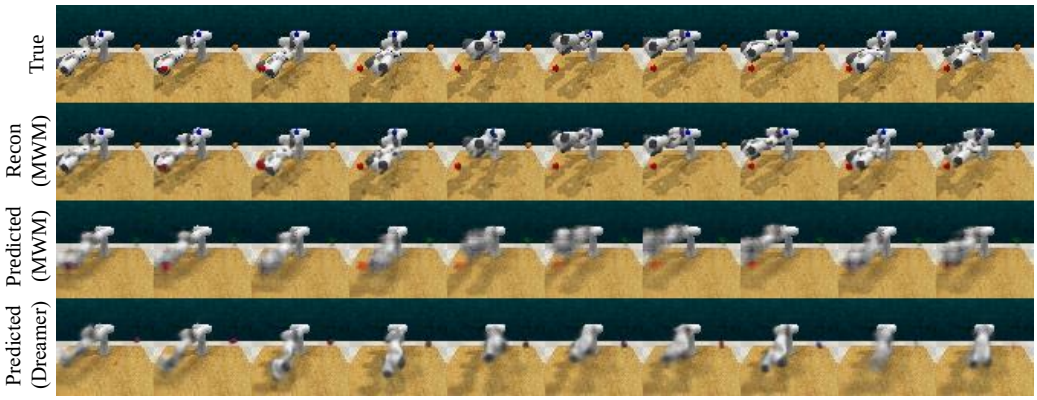


Figure 7: Future frames reconstructed with the autoencoder (*i.e.,* Recon) and predicted by latent dynamics models (*i.e.,* Predicted). Predictions from our model capture the position of a red block, which is a target position a robot arm should reach, but predictions from Dreamer are not capturing such details. In our predictions, the components that are not task-relevant are abstracted away (*i.e.,* blue and orange blocks), though the autoencoder reconstructs them. This shows how our decoupling approach works: it encourages the autoencoder to capture all the details, and the dynamics model to focus on modeling task-relevant components. Best viewed as video provided in Appendix B.

5.5 Qualitative Analysis

We visually investigate how our world model works compared to the world model of DreamerV2. Specifically, we visualize the future frames predicted by latent dynamics models on Reach Target from RLBench in Figure 7. In this task, a robot arm should reach a target position specified by a red block in visual observations (see Figure 2(c)), which changes every trial. Thus it is crucial for the model to accurately predict the position of red blocks for solving the tasks. We find that our world model effectively captures the position of red blocks, while DreamerV2 fails. Interestingly, we also observe that our latent dynamics model ignores the components that are not task-relevant such as blue and orange blocks, though the reconstructions from the autoencoder are capturing all the details. This shows how our decoupling approach works: it encourages the autoencoder to focus on learning representations capturing the details and the dynamics model to focus on modeling task-relevant components of environments. We provide more examples in Appendix B.

6 Discussion

We have presented Masked World Models (MWM), which is a visual model-based RL framework that decouples visual representation learning and dynamics learning. By learning a latent dynamics model operating in the latent space of a self-supervised ViT, we find that our approach allows for solving a variety of visual control tasks from Meta-world, RLBench, and DeepMind Control Suite.

Limitation Despite the results, there are a number of areas for improvement. As we have shown in Figure 6(c), the performance of our approach heavily depends on the auxiliary reward prediction task. This might be because our autoencoder is not learning temporal information, which is crucial for learning task-relevant information. It would be interesting to investigate the performance of video representation learning with ViTs [35, 50]. It would also be interesting to study introducing auxiliary prediction for other modalities, such as audio. Another weakness is that our model operates only on RGB pixels from a single camera viewpoint; we look forward to a future work that incorporates different input modalities such as proprioceptive states and point clouds, building on top of the recent multi-modal learning approaches [51, 52]. Finally, our approach trains behaviors from scratch, which makes it still too sample-inefficient to be used in real-world scenarios. Leveraging a small number of demonstrations, incorporating the action mode with path planning [46], or pre-training a world model on video datasets [53] are directions we hope to investigate in future works.

References

[1] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in neural information processing systems*, 2018.

[2] M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, 2011.

[3] I. Lenz, R. A. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, 2015.

[4] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

[5] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 2019.

[6] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[7] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[8] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, 2015.

[9] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

[10] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, 2019.

[11] D. Ha and J. Schmidhuber. World models. In *Advances in Neural Information Processing Systems*, 2018.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[13] 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 *International Conference on Learning Representaitons*, 2021.

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

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

[16] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[17] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020.

[18] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

[19] H. Touvron, M. Cord, A. El-Nouby, J. Verbeek, and H. Jégou. Three things everyone should know about vision transformers. *arXiv preprint arXiv:2203.09795*, 2022.

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

[21] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[22] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[23] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 2021.

[24] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. In *International Conference on Learning Representations*, 2019.

[25] T. Seyde, W. Schwarting, S. Karaman, and D. Rus. Learning to plan optimistically: Uncertainty-guided deep exploration via latent model ensembles. In *Conference on Robot Learning*, 2021.

[26] O. Rybkin, C. Zhu, A. Nagabandi, K. Daniilidis, I. Mordatch, and S. Levine. Model-based reinforcement learning via latent-space collocation. In *International Conference on Machine Learning*, 2021.

[27] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019.

[28] T. D. Nguyen, R. Shu, T. Pham, H. Bui, and S. Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, 2021.

[29] M. Okada and T. Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[30] F. Deng, I. Jang, and S. Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. *arXiv preprint arXiv:2110.14565*, 2021.

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

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

[33] G. Hinton, O. Vinyals, J. Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[34] Z. Li, Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang, et al. Mst: Masked self-supervised transformer for visual representation. In *Advances in Neural Information Processing Systems*, 2021.

[35] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022.

[36] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

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

[39] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems*, 2021.

[40] Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020.

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

[42] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, 2018.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[44] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[45] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *International Conference on Intelligent Robots and Systems*, 2012.

[46] S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 2022.

[47] S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-Fine Q-attention: Efficient learning for visual robotic manipulation via discretisation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[48] S. James and P. Abbeel. Coarse-to-Fine Q-attention with Learned Path Ranking. *arXiv preprint arXiv:2204.01571*, 2022.

[49] S. James and P. Abbeel. Coarse-to-Fine Q-attention with Tree Expansion. *arXiv preprint arXiv:2204.12471*, 2022.

[50] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[51] X. Geng, H. Liu, L. Lee, D. Schuurams, S. Levine, and P. Abbeel. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*, 2022.

[52] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv preprint arXiv:2204.01678*, 2022.

[53] Y. Seo, K. Lee, S. James, and P. Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, 2022.

[54] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[55] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.