Complex & Intelligent Systems (2024) 10:811–823
https://doi.org/10.1007/s40747-023-01174-5

ORIGINAL ARTICLE

Learning high-level robotic manipulation actions with visual predictive model

Anji Ma[1] · Guoyi Chi[2] · Serena Ivaldi[3] · Lipeng Chen[4]

Received: 13 October 2022 / Accepted: 3 June 2023 / Published online: 8 August 2023

Abstract

Learning visual predictive models has great potential for real-world robot manipulations. Visual predictive models serve as a model of real-world dynamics to comprehend the interactions between the robot and objects. However, prior works in the literature have focused mainly on low-level elementary robot actions, which typically result in lengthy, inefficient, and highly complex robot manipulation. In contrast, humans usually employ top–down thinking of high-level actions rather than bottom–up stacking of low-level ones. To address this limitation, we present a novel formulation for robot manipulation that can be accomplished by pick-and-place, a commonly applied high-level robot action, through grasping. We propose a novel visual predictive model that combines an action decomposer and a video prediction network to learn the intrinsic semantic information of high-level actions. Experiments show that our model can accurately predict the object dynamics (i.e., the object movements under robot manipulation) while trained directly on observations of high-level pick-and-place actions. We also demonstrate that, together with a sampling-based planner, our model achieves a higher success rate using high-level actions on a variety of real robot manipulation tasks.

Keywords Robot manipulation · Visual foresight · Visual perception · Deep learning · Grasp planning

Introduction

Humans can master increasingly complex manipulative behaviors and gradually develop advanced manipulation skills of exploiting high-level actions beyond lengthy primitive explorations. For example, in the infancy cycle, we typically go from only being able to fiddle with a toy to learning to grasp it directly. Then, as children, combining both high-level and low-level actions, such as push, pull, and grasp, we can accomplish more practical, efficient, and goal-oriented manipulation tasks, such as sorting a toy box. Humans naturally obtain such high-level manipulation skills through constantly observing, learning, and reproducing from interacting with the real world. It would be exciting for robots to learn to interact with objects like humans, particularly learning to incorporate high-level actions in manipulating objects with limited prior knowledge. However, learning robot manipulation skills in a real-world environment, particularly for both low-level and high-level actions, is indeed challenging.

Recently, visual foresight [12] has been widely demonstrated as a promising tool for learning visual-based robot manipulations in unknown environments from the standpoint of sensory prediction. More concretely, this line of work [7, 9, 39] is mainly built on a deep visual predictive model trained with high-dimensional visual streams for learning real-world dynamics. The learning of visual predictive models is typically task-independent [20, 31] and, therefore, can be generalized over different tasks. Even though promising results have been achieved, whether in the very vanilla visual

✉ Lipeng Chen
sclc@leeds.ac.uk

Anji Ma
maanji1993@outlook.com

Guoyi Chi
chig0002@e.ntu.edu.sg

Serena Ivaldi
serena.ivaldi@inria.fr

[1] School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

[3] Inria, University of Lorraine, CNRS, Loria, 54000 Villers-lés-Nancy, France

[4] School of Computing, University of Leeds, Leeds, UK

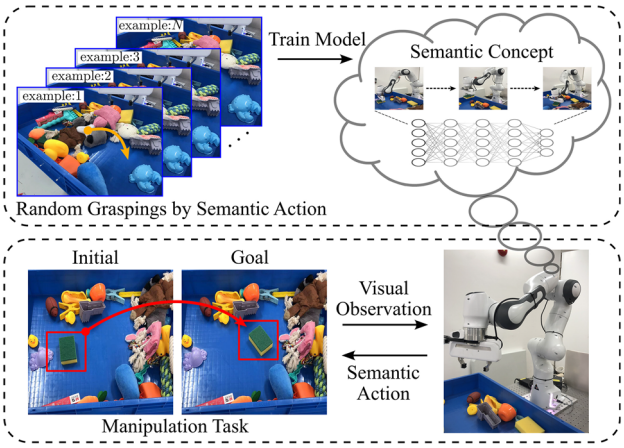

Fig. 1 Manipulation planning with visual inputs and high-level actions. Top: during the training, a visual predictive model is trained on a dataset of executed pick-and-place actions (e.g., grasping a toy to a specific location). Bottom: using the visual predictive model to optimize appropriate actions for the specific manipulation tasks (e.g., replacing a sponge's position in the workspace)

foresight [12] or its follow-up works [7, 9, 39], robot actions in this paradigm are usually prescribed to be low level, such as small differential displacements of the robot end-effector. Take the example in Fig. 1. Even to relocate a sponge within a quite short distance, a robot using visual foresight will typically have to apply a fairly long sequence of low-level displacements, while a human expert can come up with more efficient solutions with higher level manipulation actions, such as simply picking (grasping) the sponge up and placing it to the target location directly. Moreover, such low-level actions are babbling-like, not only in the training data, e.g., RoboNet [3], but also in the short horizon of actions planned through the learned predictive model using model predictive control (MPC). Such a planning framework is usually confronted with increasing complexity for tasks requiring long sequences of low-level displacements of the robot end-effector.

To address these limitations, we go deeper into visual foresight by improving the model's understanding of high-level actions in robot manipulations. The underlying intuition is to train a deep visual predictive model that can learn world dynamics under high-level robot actions and ultimately use it to determine appropriate high-level actions in task planning. However, training such a predictive model remains a challenge. High-level actions usually contain rich semantic information and cues that are not present in low-level actions, which poses two questions regarding understanding such actions. First, how can the robot learn a visual predictive model by leveraging the semantic information in high-level actions? Second, how can the robot learn high-level actions while still retaining the understanding of low-level actions? Learning semantic information from low-dimensional task representations, such as the object bounding box and semantic segmentation [34], usually relies on ground true annotation, which can hardly be achieved while learning from a large number of raw visual observations. Instead, we incorporate semantic information into a sequence of visual frames obtained under high-level robot actions and build a recurrent neural network to learn such information implicitly. This allows our model to learn both high-level and low-level robot actions from the interval of consecutive frames. The main contributions of this paper are summed up as follows:

- We propose a novel visual predictive model for high-level robot actions containing an action decomposer and a video prediction model.
- We present a sampling-based optimization method to utilize this visual predictive model for planning high-level pick-and-place actions in real robot tasks.
- We contribute a novel vision dataset that contains a rich set of real robot pick-and-place actions.

We evaluate our method in terms of the accuracy of the predicted outcomes of high-level actions and the overall performance of using the predictive model in real robot downstream tasks. The results demonstrate that our approach can substantially learn to understand high-level robot actions and can promisingly be utilized in planning for real robot manipulation tasks. A video summary of this paper and more experimental results can be found at https://youtu.be/JOgjovETlVg.

Related work

Model-based reinforcement learning

The main difference between model-based reinforcement learning (RL) and model-free RL is the employment of world models learning transition dynamics in model-based RL. Model-based methods usually have more data efficiency than model-free methods [6] and require fewer reward signals during training. These can significantly reduce the robot–environment interaction in learning, which is often expensive and dangerous for robots. Model-based RL in robotics [5, 26] has attracted many studies in the last decade and has shown great success in low-dimensional environments [15, 17]. Recently, a line of literature called visual foresight [7, 9, 12] has proposed a way that leverages raw visuals directly in the model-based context. In visual foresight [12], a predictive model is trained to learn the concepts of robot actions by accurately predicting the visual outcomes based on both current visual observations and robot actions. Furthermore, the predictive model is task-agnostic, allowing it to be generalized over various tasks. Such an approach has shown

robustness that can process real-world visual inputs and has demonstrated promising performance in real robot tasks. However, the focus of vanilla visual foresight [12] and its follow-up works [7, 9] is to leverage only low-level robot actions in prediction and planning. In contrast, our method learns a visual predictive model conditional on higher level robot actions, which can be used for more complex and efficient action planning. Hafner et al. [20] proposed a model-based approach that learned dynamics directly from pixels but plans actions in a latent space. Their approach has shown great success on a larger scale of longer horizon tasks in simulated environments. However, this method still requires some labeled data for training a reward function. In comparison, we focus on learning robot actions from only raw visual streams rich in real-world visual complexity.

Video prediction model

Recently, deep neural networks have made great progress in representing high-dimensional states and observations. Video prediction models have become a powerful tool for learning world dynamics in various domains, including autonomous driving [16, 29], human posture estimations [8, 23], and robotic manipulation [10]. These models learn from a large amount of unlabeled data in a self-supervised manner by utilizing the vision as a supervised signal. From earlier deterministic models [2, 11, 35] to VAE-based [24] models [1, 7, 25, 40], a latent space is employed in probabilistic models to catch the stochasticity of the real environment. To model time-variational stochasticity, [7] proposed using a learned prior in the stochastic model. The action-conditional video prediction models have been used to learn the robot's actions, making them suited to the robot context.

Robot manipulations

Pick-and-place is a wide-spread action of robot manipulation in various robotics applications, including industrial [41] and domestic applications [37]. Traditionally, this problem has been studied through analytical estimation of object poses [32] and dynamic motion planning [13]. Both require object models and are unsuitable for unstructured environments. In recent years, data-driven methods for learning pick-and-place actions have gained significant attention in robotics, with both model-free [18] and model-based [14] techniques. Several works use learned geometric models [38, 44] to estimate object poses and infer actions. However, these methods still require object models during training. End-to-end models [33, 42] have the advantage of being agnostic to the object's physics. They can directly infer the pick-and-place actions from pixels. An instance is the Transporter network [42], which utilizes a simple model architecture that can exploit spatial symmetries to effectively learn to plan pick-and-place actions from visual inputs. However, this line of methods tends to be task-specific and relies on task-specific demonstrations, which limits them from zero-shot generalization to new tasks.

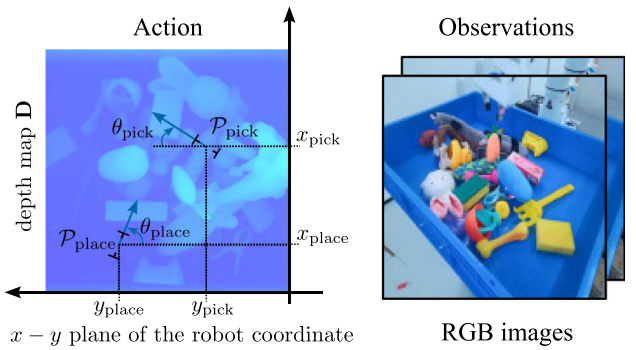

Fig. 2 Left: each pick-and-place action is formulated in SE(2) on the xy plane of the robot workspace, and the coordinate z is determined by the depth w.r.t. the plane. Right: the visual observations are acquired from an RGB camera above the workspace

Another sub-field of the pick-and-place actions is predicting the probability of success of picking. For instance, Dex-Net [28] uses a grasp quality convolutional neural network (GQ-CNN) to estimate optimal picking poses from a depth image. However, it does not consider task-related objectives such as which object should be picked and where it should be placed.

Problem formulation

In this work, our objective is to learn high-level robot actions through a visual predictive model and ultimately enable their use in robot manipulation planning. To this end, we formulate the completion of a manipulation task by one or a sequence of high-level actions. We define each high-level action as a pick-and-place action grasping an object from above a *pick* position and releasing the gripper at a *place* position. The poses of the robot gripper at the pick and the place are both composed of a 3-D coordinate x, y, z, and a yaw rotation θ in the robot base coordinate. As shown in Fig. 2 (left), we parameterize the high-level action as $\mathbf{a}^{(high)} = (\mathcal{P}_{\text{pick}}, \mathcal{P}_{\text{place}}) \in \mathcal{A}$, where $\mathcal{P}_{\text{pick}}$ and $\mathcal{P}_{\text{place}}$ are the poses of picking and placing defined in SE(2), which refers to the xy plane of the robot base coordinate system. z is determined as the vertical depth w.r.t. the horizontal plane. An RGB camera is used to acquire the visual observations of the workspace [Fig. 2 (right)].

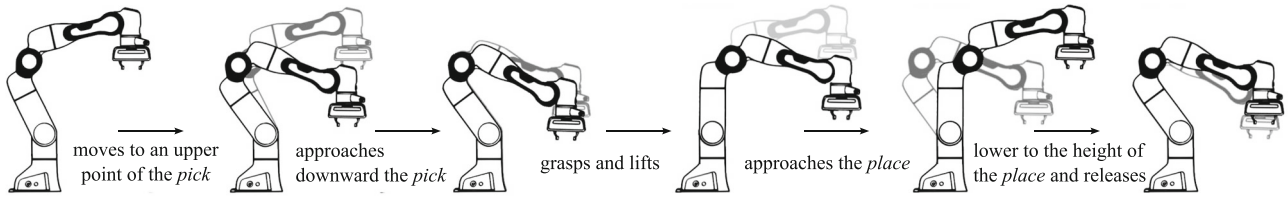

Fig. 3 The sequential motion primitives of a pick-and-place action: the robot moves the gripper to the upper point of the pick point, approaches the object down to the height of the pick point, grasps the object with the gripper, lifts the gripper back to the upper point, approaches the place location, lowers the gripper to the height of the place location, and finally releases the gripper

Method

This section describes our approach to learning high-level robot actions by integrating a decomposer with a video prediction model and applying the learned model using sampling-based optimization techniques for desired tasks. An illustration of our method is shown in Fig. 5.

Visual predictive model of high-level actions

We use the notation $\mathcal{M} : \{\mathbf{I}_{\text{init}}, \mathbf{a}^{(high)}\} \rightarrow \hat{\mathbf{I}}$ to refer to a visual predictive model, where $\mathbf{I}_{\text{init}}$ is the initial visual observation and $\hat{\mathbf{I}}$ is the predicted visual outcome of a pick-and-place action $\mathbf{a}^{(high)} = (\mathcal{P}_{\text{pick}}, \mathcal{P}_{\text{place}})$. Model $\mathcal{M}$ learns to understand high-level robot actions by being trained to predict visual outcomes. It is worth emphasizing that high-level actions like pick-and-place contain semantic information. For example, as shown in Fig. 3, the robot executes a pick-and-place action through several semantic steps: the robot moves the gripper to an upper point of the pick location, approaches the object down to the height of the pick, executes the gripper to grasp, lifts the gripper back to an upper point, approaches the place location, lowers the gripper to the height of the place, and releases the gripper. However, such semantic information can not be incorporated by the action formulation $\mathcal{P}_{\text{pick}}$ and $\mathcal{P}_{\text{place}}$.

To leverage such semantic information in the prediction, we propose combining a video prediction model with a high-level action decomposer. The decomposer converts a high-level action into a sequence of intermediate low-level actions. We thus incorporate the semantic information through the resulting intermediate visual frames and low-level actions. In the literature on robot manipulation [9], video prediction models typically predict visual frames autoregressively conditional on a sequence of low-level actions, i.e., the displacements of the robot end-effector. The advantages of using these two components together are twofold: (1) the semantic information is still retained in the decomposed low-level action sequence; (2) the model can learn both low-level and high-level actions concurrently.

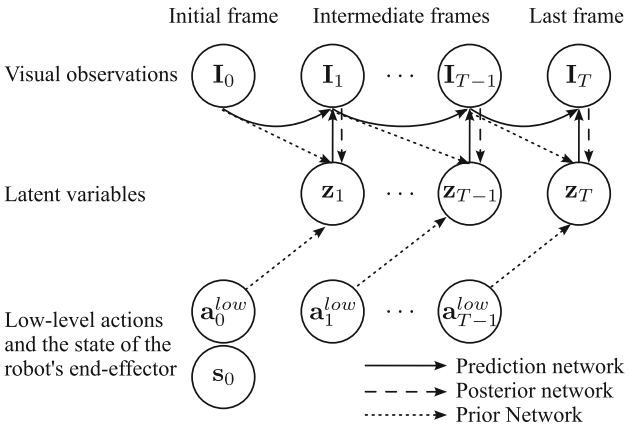

Fig. 4 A probabilistic video prediction model conditional on robot actions. High-level action is decomposed into a sequence of low-level displacements $\mathbf{a}_0^{low}, \mathbf{a}_1^{low}, \ldots \mathbf{a}_{T-1}^{low}$ and an initial position $\mathbf{s}_0$ of the robot's end-effector

Specifically, we decompose the high-level action into a sequence of low-level actions with $\mathbf{a}^{(high)} \rightarrow \{\mathbf{s}_0, \mathbf{a}_0^{(low)}, \mathbf{a}_1^{(low)}, \ldots, \mathbf{a}_{T-1}^{(low)}\}$, where $\mathbf{s}_0$ denotes the robot's initial state that includes the end-effector's pose (x, y, z, θ) and a binary scalar (open v.s. closed) of the gripper. $\mathbf{a}_t^{(low)}$ is the intermediate low-level action between two successive frames at time t, denoting the end-effector's displacement $(\Delta x, \Delta y, \Delta z, \Delta\theta)$ and the binary scalar of the gripper. $\mathbf{I}_0$ is the initial frame and $\mathbf{I}_t$ is the resulting frame of action $\mathbf{a}_{t-1}^{(low)}$. T is the length of the sequence of low-level actions.

Figure 4 shows a schematic of the video prediction model. At each step t, the model takes the observation $\mathbf{I}_t$ and action $\mathbf{a}_t^{(low)}$ as input and generates the next predicted frame $\hat{\mathbf{I}}_{t+1}$. By performing this prediction procedure autoregressively, i.e., using the predicted frame $\hat{\mathbf{I}}_{t+1}$ as the input for the next time step, we can predict the last frame of the low-level action sequence conditional on an initial frame.

To train the model, we gather a dataset $\mathcal{D} = \{\xi_i\}_{i=1}^N$ of N high-level robot of actions, where each example ξ_i contains of a pick-and-place action $(\mathcal{P}_{\text{pick}}, \mathcal{P}_{\text{place}})$, its low-level decomposition $\{\mathbf{s}_0, \mathbf{a}_0^{(low)}, \mathbf{a}_1^{(low)}, \ldots, \mathbf{a}_{T-1}^{(low)}\}$, and the corresponding visual frames $\{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_T\}$.

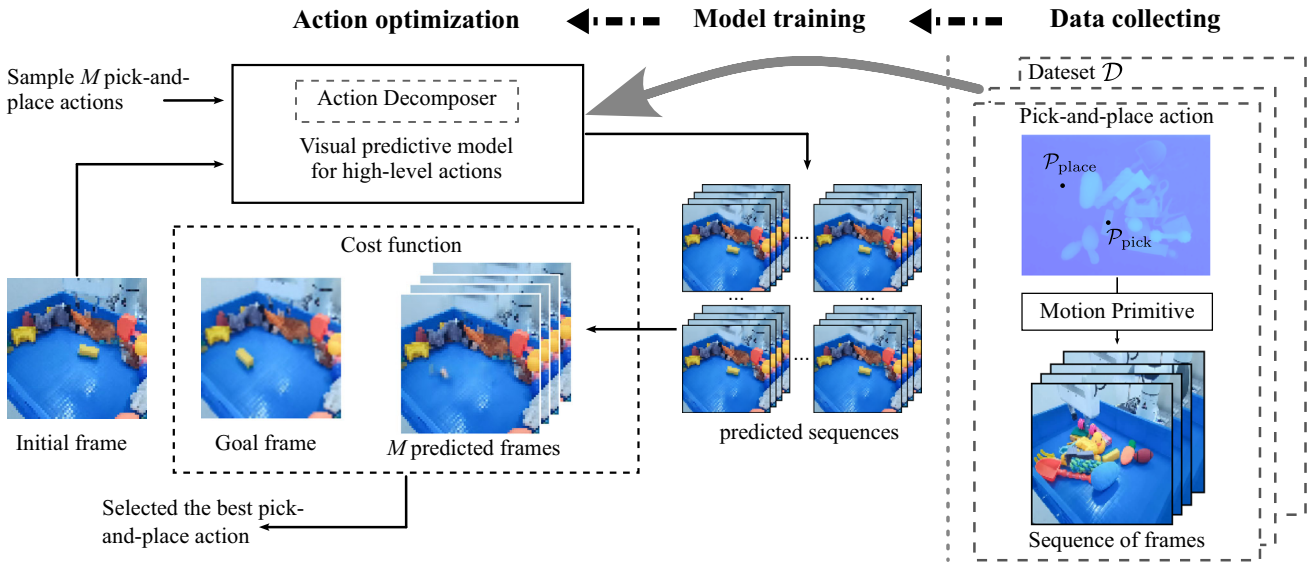

Fig. 5 An overview of our method, including training a visual predictive model for high-level pick-and-place actions and using it with sample-based optimization for action planning

Variational video prediction model

Variational auto-encoders (VAE) [24] have been widely used for video prediction models. Following the action conditional video prediction paradigm, the prediction models typically take c initial frames $\{\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_{c-1}\}$ and a sequence of action $\{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\}$ as inputs and predict subsequent future frames $\{\mathbf{I}_c, \mathbf{I}_{c+1} \ldots, \mathbf{I}_T\}$. VAEs introduce latent variables $\mathbf{z} \sim p(\mathbf{z})$ to carry the stochastic nature of the real world. Thus, we can build a probabilistic model $p_\theta(\mathbf{I}_t|\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}, \mathbf{z}_{1:t})$ that predicts the frame $\hat{\mathbf{I}}_t$ conditioned on the previous frames $\mathbf{I}_{0:t-1}$, actions $\mathbf{a}_{0:t-1}$ and the latent variables $\mathbf{z}_{1:t}$. Since estimating the marginalized distribution over the latent space $\mathbf{z}$ is intractable, it is not possible to directly maximize $p_\theta(\mathbf{I}_t)$. To overcome this challenge, VAEs employ an inference network $q_\phi(\mathbf{z}_t|\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1})$ to approximate the posterior of the true distribution of the latent variables $\mathbf{z}$. This posterior inference network is typically parameterized as a conditional Gaussian distribution $\mathcal{N}(\mu_\phi(\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}), \sigma_\phi(\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}))$.

By utilizing the reparameterization strategy [24]

$$\mathbf{z} = \mu_\phi(\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}) + \sigma_\phi(\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}) \times \epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}), \tag{1}$$

the model can be trained by optimizing the variational lower bound of the log-likelihood

$$\mathcal{L}_{\theta,\phi}(\mathbf{I}_{c:T}) = \sum_{t=c}^{T} \Big[\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{I}_{0:t},\mathbf{a}_{0:t-1})} \log p_\theta(\mathbf{I}_t|\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}, \mathbf{z}_{0:t}) - \beta D_{KL}(q_\phi(\mathbf{z}_t|\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}) \| p(\mathbf{z}_t))\Big]. \tag{2}$$

To capture the variety of the stochastic information, Denton et al. [7] propose a learned-prior $p_\psi(\mathbf{z}_t|\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1})$. This prior can also be parameterized as a conditional Gaussian distribution $\mathcal{N}(\mu_\psi(\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}), \sigma_\psi(\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}))$.

The complete model is trained by maximizing

$$\mathcal{L}_{\theta,\phi,\psi}(\mathbf{I}_{c:T}) = \sum_{t=c}^{T} \Big[\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{I}_{0:t},\mathbf{a}_{0:t-1})} \log p_\theta(\mathbf{I}_t|\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}, \mathbf{z}_{0:t}) - \beta D_{KL}(q_\phi(\mathbf{z}_t|\mathbf{I}_{0:t}, \mathbf{a}_{0:t-1}) \| p_\psi(\mathbf{z}_t|\mathbf{I}_{0:t-1}, \mathbf{a}_{0:t-1}))\Big], \tag{3}$$

where θ, ϕ, and ψ are the parameters of the generative network, posterior network, and prior network, respectively. D_{KL} is the Kullback–Leibler divergence between the approximated posterior and the learned prior. β is a hyperparameter representing the trade-off between minimizing the prediction error and fitting the prior. During training, the latent variables $\mathbf{z}_t$ are sampled from the posterior $q_\phi(\mathbf{z}_t)$. During testing, we directly sample $\mathbf{z}_t$ from the learned-prior $p_\psi(\mathbf{z}_t)$.

The implementation of this model contains an encoder, a decoder, a prior network, a prediction network, and a posterior network. Both the encoder and decoder are deep convolutional neural networks that map the pixels to latent space and map it back to the pixels, respectively. The prior, prediction, and posterior networks are convolutional LSTM networks for learning long-term dependencies.

Action planner with visual predictive model

The objective of a robot manipulation planner is to find one or a sequence of pick-and-place action(s) that maximizes the

Algorithm 1: Planning with the prediction of pick-and-place actions

Input: Visual predictive model $\mathcal{M}$,
Cost function $C = \ell_1$,
Goal frame $\mathbf{I}_{\text{goal}}$

Initialize $n_{\text{step}} = 0$
repeat
  Obtain an initial observation frame $\mathbf{I}_{\text{init}}$
  Initialize a multivariate Gaussian distribution
  for $i \leftarrow i$ **to** n_{iter} **do**
    Sample M actions $\{\mathcal{P}^{(m)}_{\text{pick}}, \mathcal{P}^{(m)}_{\text{place}}\}^M$ from the multivariate Gaussian distribution.
    Use $\mathcal{M}$ to predict $\{\hat{\mathbf{I}}^{(m)}_{1:T}\}^M$ conditioned on sampled actions.
    Evaluate each action through the cost function: $c^{(m)} = 1 - C(\hat{\mathbf{I}}^{(m)}_T, \mathbf{I}_{\text{goal}})$.
    Fit the multivariate Gaussian distribution to the K actions with the lowest cost.
  end
  Execute the action $\{\mathcal{P}^*_{\text{pick}}, \mathcal{P}^*_{\text{place}}\}$ with the lowest cost.
until *SUCCESS or* $n_{step} = max_{step}$;

possibility of achieving the given goal frame $\mathbf{I}_{\text{goal}}$. We evaluate a pick-and-place action by computing the ℓ_1 cost function between the predicted frame of it and the goal frame. Subsequently, we optimize the pick-and-place actions using a sample-based algorithm known as the cross-entropy method (CEM).

The procedure is shown in Algorithm 1. Concretely, for an initial frame $\mathbf{I}_{\text{init}}$ and a goal frame $\mathbf{I}_{\text{goal}}$, we sample M pick-and-place actions $\{\mathcal{P}^{(m)}_{\text{pick}}, \mathcal{P}^{(m)}_{\text{place}}\}^M$ from a normal multivariate Gaussian distribution. We predict the visual outcomes $\hat{\mathbf{I}}^{(m)}_{1:T}$ for each pick-and-place action, and then evaluate the cost of each action using $c^{(m)} = \ell_1(\hat{\mathbf{I}}^{(m)}_T, \mathbf{I}_{\text{goal}})$. We then select K actions with the lowest costs, fit a new multivariate Gaussian distribution on these K pick-and-place actions, and resample a new set of M actions from this new distribution. We repeat the prediction and refitting procedures for n iterations. After the final iteration, we execute the pick-and-place action $\{\mathcal{P}^*_{\text{pick}}, \mathcal{P}^*_{\text{place}}\}$ with the lowest cost, which has the predicted visual outcome closest to the given goal.

Since some complex tasks may require more than one pick-and-place action, we adopt a greedy strategy that selects the action with the lowest cost at each step and optimizes it anew over a current frame until the task is successful within the maximum steps. In contrast to previous approaches that used CEM and MPC to optimize low-level actions, our method optimizes high-level actions, resulting in greater planning efficiency. By avoiding the need for repeated optimization at each step of a low-level action, our approach is more closely aligned with human planning strategies that involve selecting actions from a higher level.

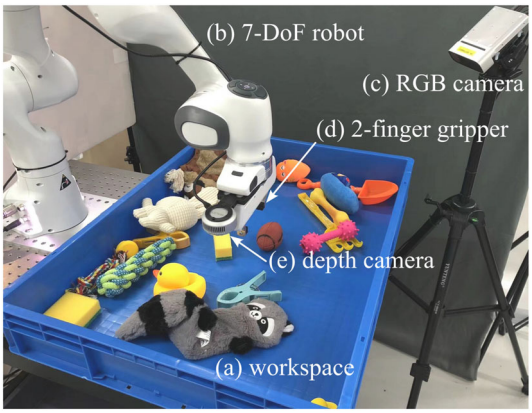

Fig. 6 Environmental setup. It includes **a** a horizontal workspace, **b** a 7-DoF Franka Emika Panda robot, **c** an RGB camera observing the workspace, **d** a 2-finger Franka Emika gripper, and **e** a depth camera used to obtain the height position of each action

Experiments and results

Our experiments aim to evaluate whether the robot can learn high-level actions through the proposed visual predictive model and ultimately leverage them in real-world robot manipulation tasks. The question is twofold: (1) Can the proposed method predict accurate visual outcomes of robot pick-and-place actions? and (2) can the visual predictive model learning pick-and-place actions be used to plan real robot tasks, and can this lead to a greater success rate and planning efficiency?

We conduct both quantitative and qualitative experiments to answer the above questions. To answer question (1), we compare the predicted visual outcomes of pick-and-place actions between our method and baseline methods that use either a conditional variational autoencoder (CVAE) or models trained only on data of low-level actions. For question (2), we compare our approach with the vanilla visual MPC that uses only low-level robot actions on a variety of real robot tasks. In addition, we further conduct experimental comparisons with other pick-and-place methods, including Transporter Network [30, 43] and Dex-Net [28], where custom algorithms are designed, but no world dynamic models are considered. More visualizations and videos can be found at https://youtu.be/JOgjovETlVg.

Experimental setup

We train and evaluate our proposed method in a real robot environment. For both data collection and evaluation, we use a 7-DoF Franka Emika Panda robot equipped with a two-finger Franka Emika gripper, as shown in Fig. 6. To obtain visual observations, we place an RGB camera on the side and process the images to a 64x64 pixel resolution for the predictive model. Since we parameterize pick-and-place actions on

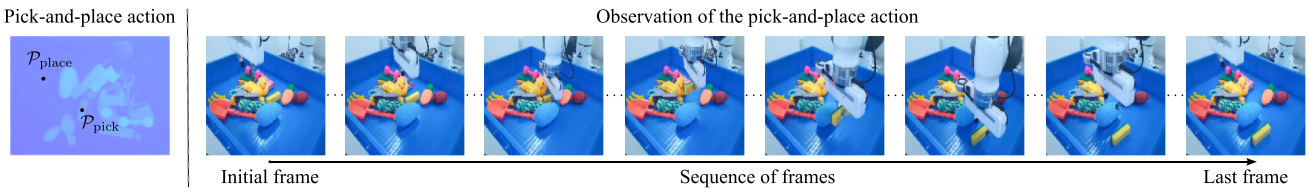

Fig. 7 An example in PandaGrasp-Pick&Place dataset. Left: the pick-and-place action. Right: the observation of a sequence of visual frames

a horizontal 2D plane in SE(2), another depth camera providing the depth map of the workspace is mounted at the robot's end-effector to obtain the height position (z-axis) of each action. In our experiments, all models are trained with 4× NVIDIA Tesla V100 (32 GB) graphics cards, while inferences are done with one consumer graphics card NVIDIA GeForce 3090.

Dataset

We collect a real robot dataset named PandaGrasp-Pick&Place, to train and evaluate the proposed method. While RoboNet [4] introduced an autonomous data collection strategy to obtain data on interactions between the robot and objects in open-world environments and released a promising dataset, actions in RoboNet are mostly low-level displacements of the robot's end-effector. To address this limitation, we introduce high-level pick-and-place action on our dataset. Specifically.

RoboNet

[4] provides a large open dataset containing 150K trajectories of robot manipulation from several robots. Each trajectory in RoboNet records a sequence of visual observations and low-level actions defined as the displacements of the robot end-effector. Despite RoboNet providing a large number of examples, the babbling-like exploration strategy results in a scarcity of high-level actions in the provided examples. In our experiments, we use RoboNet to pre-train the visual predictive model of our method and establish baselines for evaluating the visual prediction performance.

PandaGrasp-Pick&Place

As its name indicates, it is a dataset containing the pick-and-place actions of a Franka Panda robot. Concretely, as shown in Fig. 7, each example in the dataset records the robot performing a random pick-and-place action. The robot executes the actions according to the primitives defined in Fig. 3, which involve approaching the picking position, grasping the object, and moving to the placing position. We record the visual observation of each action as a sequence of 21 frames in length, according to the high-level action decomposer proposed in the section "Method".

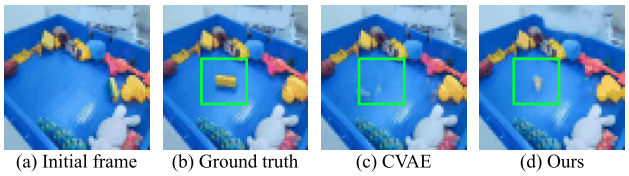

Fig. 8 Qualitative comparison between the predictions of a CVAE network and our proposed method. **a** The initial frame, **b** the ground truth of the last frame, **c** the prediction of the CVAE network, and **d** the prediction of our proposed method. The green boxes on the pictures highlight that a sponge is being manipulated in this example

Table 1 Quantitative comparisons of the predicted last frames between the CVAE network and our proposed method (mean ± standard error)

Method	PSNR ↑	SSIM ↑	LPIPS ↓
CVAE	21.8 ± 0.02	87.2 ± 0.03	0.036 ± 0.001
Ours	**22.5 ± 0.02**	**87.4 ± 0.04**	**0.032 ± 0.001**

Visual prediction conditional on high-level actions

To study whether the visual predictive model can understand robot pick-and-place actions correctly, we evaluate the accuracy of the predicted frames of such actions in reference to the ground truth frames. Our quantitative evaluations are with three metrics: structural similarity index measure (SSIM) [36], peak signal-to-noise ratio (PSNR) [22], and learned perceptual image patch Similarity (LPIPS) [45].

We first compare our method with a CVAE network that directly predicts the last frame of a pick-and-place action given an initial frame. For a fair comparison, the CVAE network shares the same encoder and decoder structure as our method, and both methods are trained on the same data. The results (Table 1) show that compared with the CVAE network, using our proposed method to predict the visual outcomes of pick-and-place action achieves better performance on all metrics.

Figure 8 presents a qualitative comparison between the predictions of the CVAE network and our proposed method. The initial frame used as input to both methods is shown as Fig. 8a, b which shows the ground truth of the last frame. Figure 8c, d shows the predictions of the CVAE network and ours, respectively. The green boxes on the pictures highlight that a sponge is being manipulated in this example. Upon

Table 2 Quantitative comparison among different visual predictive models trained on high-level or low-level actions on the average of the sequence of frames (mean ± standard error)

Model	PSNR ↑	SSIM ↑	LPIPS ↓
Trained on RoboNet	18.9 ± 0.2	75.6 ± 0.3	0.065 ± 0.003
Pre-trained on RoboNet and fine-tuned on Panda-Babbling	22.0 ± 0.1	85.4 ± 0.3	0.030 ± 0.001
Pre-trained on RoboNet and fine-tuned on PandaGrasp-Pick&Place (ours)	**22.9 ± 0.2**	**86.8 ± 0.4**	**0.026 ± 0.001**

Table 3 Quantitative comparison among different visual predictive models trained on high-level or low-level actions on different stages of pick-and-place actions (mean ± standard error)

Stage	Model	Visual predictive performance (test) PSNR ↑	SSIM ↑	LPIPS ↓	Stage	Model	Visual predictive performance (test) PSNR ↑	SSIM ↑	LPIPS ↓
(1) Approach to the pick	m1	23.3 ± 0.2	88.7 ± 0.3	0.021 ± 0.003	(2) Grasp at the pick	m1	18.1 ± 0.2	74.5 ± 0.3	0.068 ± 0.003
	m2	**27.2 ± 0.1**	**93.9 ± 0.2**	0.009 ± 0.001		m2	**24.5 ± 0.1**	**91.2 ± 0.3**	**0.017 ± 0.001**
	m3 (ours)	26.2 ± 0.2	92.9 ± 0.3	**0.008 ± 0.001**		m3 (ours)	23.1 ± 0.1	88.0 ± 0.3	0.018 ± 0.001
(3) Move to the place	m1	17.9 ± 0.3	73.2 ± 0.3	0.070 ± 0.002	(4) Release the gripper	m1	17.6 ± 0.2	71.2 ± 0.4	0.083 ± 0.005
	m2	21.2 ± 0.2	85.3 ± 0.3	0.036 ± 0.001		m2	18.1 ± 0.2	77.3 ± 0.3	0.052 ± 0.002
	m3 (ours)	**22.0 ± 0.1**	**86.0 ± 0.3**	**0.029 ± 0.001**		m3 (ours)	**21.2 ± 0.2**	**83.7 ± 0.5**	**0.038 ± 0.001**

*m1: model trained on RoboNet; *m2: model pre-trained on RoboNet and fine-tuned on Panda-Babbling; *m3: model pre-trained on RoboNet and fine-tuned on PandaGrasp-Pick&Place (ours)

comparing the predictions to the initial frame, both methods predict that the sponge has been moved from its initial position. However, when we compare the predicted frames to the ground truth, only our method accurately predicts that the sponge has been moved to the intended place position. In contrast, the CVAE network fails to generate a reasonable object in the prediction results. This is because the CVAE network learns the pick-and-place actions only from mapping pixels between the initial and the last frames. Our method, however, learns pick-and-place actions through a sequence of intermediate frames, which have more semantic information about the actions. This enables us to correctly predict the sequence of frames up to the last frame we are interested in.

We also conduct a comparison with two baselines to evaluate whether or not the visual predictive model still has the ability to learn high-level actions in the absence of such actions. Specifically, the baseline models are trained using only low-level actions (i.e., the end-effector's displacements), similar to those in RoboNet. To mitigate the bias from the specific robots and environments in our dataset, we acquire a dataset with our setup but adopt the babbling-like methodology in RoboNet. We refer to this dataset as Panda-Babbling. The compared baselines and our model are as follows:

(1) A visual predictive model trained on RoboNet.
(2) A visual predictive model pre-trained on RoboNet and fine-tuned on Panda-Babbling.
(3) A visual predictive model pre-trained on RoboNet and fine-tuned on PandaGrasp-Pick&Place.

Table 2 shows the average quantitative results over the prediction of the entire sequence of pick-and-place actions. The model trained on pick-and-place action data outperforms other models trained only on low-level actions. We then evaluate the models on different stages of pick-and-place actions (Table 3). Although the model trained on Panda-Babbling performs better in the short horizon, such as the first two stages, it fails to predict in the long horizon, which is important for pick-and-place actions. In contrast, our method trained on PandaGrasp-Pick&Place performs better prediction on the long horizon.

Figure 9 shows a qualitative comparison of prediction results across different models. Although the model trained on Panda-Babbling successfully learns the gripper movements related to the low-level displacement actions, it fails to learn object movements. In contrast, the model trained on PandaGrasp-Pick&Place achieves more accurate predictions of both gripper and object movements.

Evaluation on real tasks with high-level actions

This section evaluates whether using high-level actions in the prediction and planning leads to a greater success rate in real robot tasks and more efficiency in planning, especially for tasks related to pick-and-place actions. As shown in Fig. 10, we compare our method with the vanilla visual MPC [9, 39] on three manipulation tasks, including

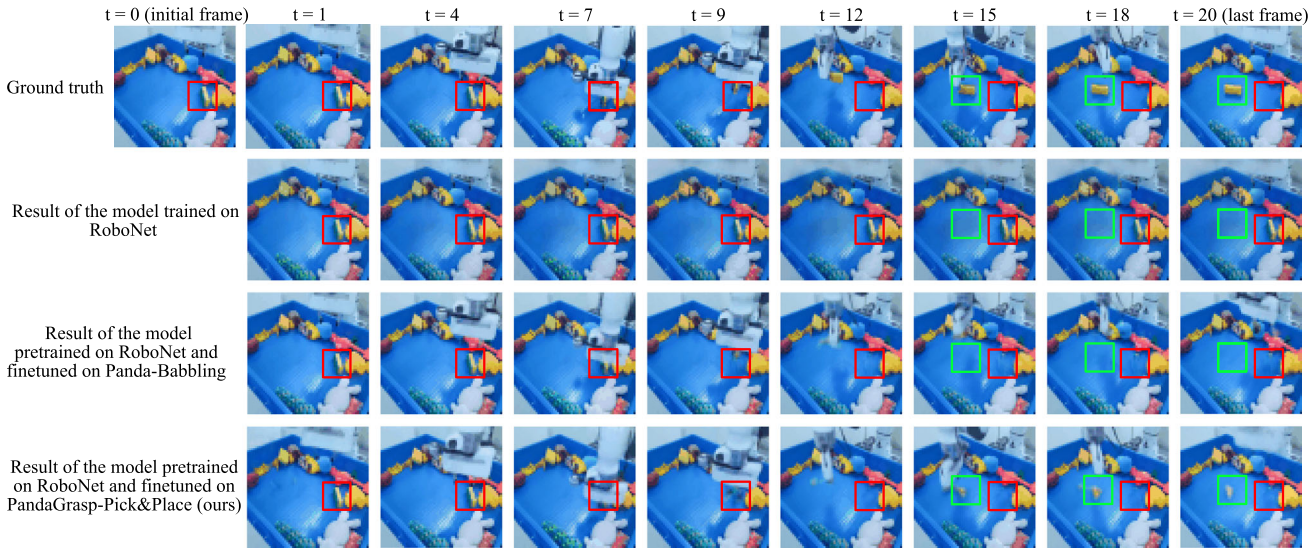

Fig. 9 Qualitative comparison among the predictions of visual predictive models that are whether being trained on the data of pick-and-place actions or not. We show some keyframes of the sequence of frames. The red boxes highlight whether the sponge is correctly predicted to be picked up, and the green boxes highlight whether the sponge is correctly predicted to be placed

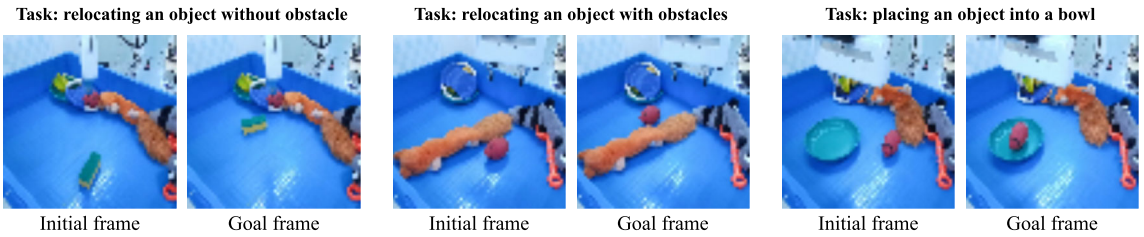

Fig. 10 Three real robot tasks in our experiments. Left: relocating an object without obstacles. Middle: relocating an object with obstacles. Right: placing an object into a bowl

1. Relocating an object without obstacles, where the goal is to relocate an object into a desired position without obstacles in the workspace.
2. Relocating an object with obstacles, where the goal is to move the object across to a new location without impacting obstacles in the workspace.
3. Placing an object into a bowl, where the robot is required to place an object into a particular target area such as a bowl.

Given an initial frame and the goal frame, our method performs visual predictions on a set of pick-and-place actions. We then select the action with the predicted frame resulting in the lowest ℓ_1 loss to the goal frame. In contrast, for the vanilla visual MPC, we adopt the planner method described in citewu2021greedy, where the predictive model predicts the frames on a set of sequences of low-level actions on a short horizon and iteratively selects the first action of the sequence that leads to the lowest ℓ_1 loss to the provided goal frame. For each task, we repeat the experiments with ten configurations by randomly putting objects in the workspace and designating a goal visual frame according to the corresponding task specifications. Table 4 lists the various objects in our experiments to diversify the configurations.

Table 4 Various objects used in our experiments

Objects	Sponge (face up/ back side up), toy football
Obstacles	Plush toy, toy hammer
Bowl	Round, rectangle bowls

We annotate an experiment as a success if the target object is relocated or placed into the goal configuration within the maximum number of steps and as a failure otherwise. In Fig. 11, we present a set of qualitative experiments showing that the learned model of high-level actions can complete all three manipulation tasks related to pick-and-place actions. Table 5 shows that using high-level action prediction and planning leads to higher success rates than vanilla visual MPC. This is particularly apparent in tasks that involve more complex robot and/or object interactions, e.g., with obstacles or other objects in the scene.

Table 5 Comparisons between success rate and efficiency of different planning methods

	Relocating	Relocating with obstacles	Placing into a bowl	Avg. iters
Vanilla visual MPC	**70%**	10%	30%	9.3
Planning on high-level action prediction (ours)	60%	**50%**	**60%**	1.2 (7.7×)

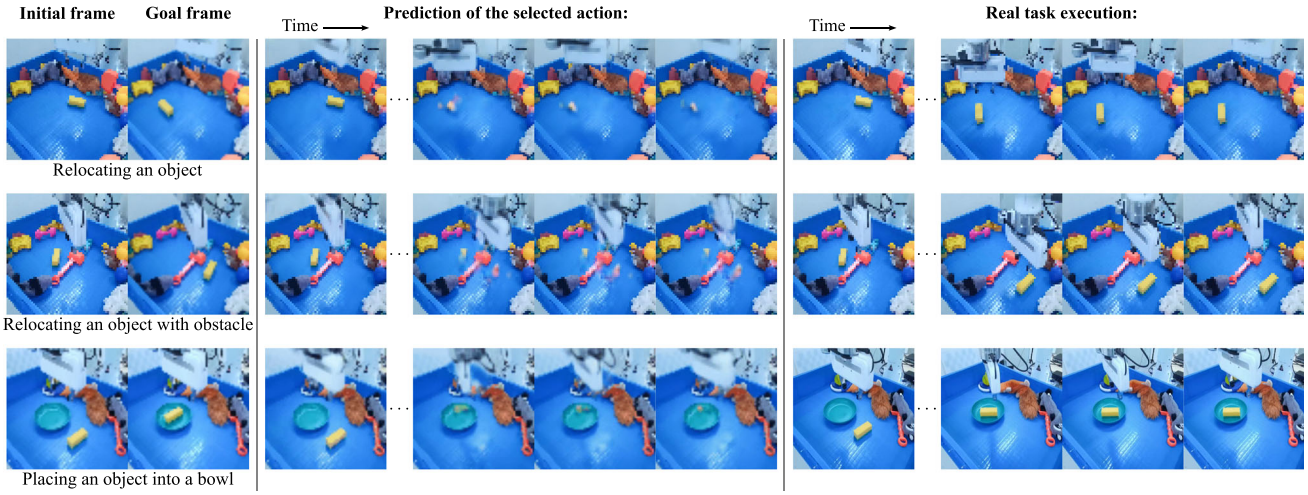

Fig. 11 Qualitative visualization for using the prediction of pick-and-place actions in three real robot tasks. Left: the given initial and goal frames of a specific task. Middle: the prediction of the planned action. Right: the real execution of the robot

Furthermore, Table 5 shows that the average number of CEM iterations of planning on pick-and-place actions is much less (7.7×) than that of planning on low-level actions. This highlights that planning on high-level pick-and-place actions leads to greater efficiency in downstream tasks. Regardless of our method or vanilla visual MPC, each CEM iteration denotes an action re-planning process. In vanilla visual MPC, action re-planning occurs after every low-level action. In contrast, our method only performs the re-planning after the high-level action, resulting in greater planning efficiency. Although the autoregressive generation makes our method predict a longer horizon than vanilla visual MPC (20 frames vs. ten frames), planning pick-and-place actions allows for a reduction in the number of CEM iterations, still resulting ins greater planning efficiency.

Comparison with other planning frameworks

In this section, we provide experimental analysis and compare our approach with other state-of-the-art planning frameworks of pick-and-place actions, such as the Transporter Network [30, 43], and Dex-Net [28].

Transporter network

The Transporter network leverages visual cues to determine the task's goal and ultimately uses them to estimate the robot's pick-and-place actions. To compare with the Transport network, we replicate it to perform robot rearranging tasks in our local environment. Following [43], we implement a similar user interface to acquire human demonstrations, which we use to train the model of the Transport network.

Concretely, we obtain 500 human demonstrations of placing an object into a round bowl, as shown in the left part of Fig. 12. The results in Table 6 demonstrate that we have trained a model that performs very well (a 100% success rate) on the task of "placing an object into a round bowl". However, we also observe a limitation in using visual cues to generalize between tasks. Table 6 shows that this model performs very limitedly on the new task of "placing an object into a rectangle bowl". This is due to the learning of the Transport network being task-specific, and the demonstrations used for training limit the model to only manipulate with a round bowl. In contrast, for our approach, we aim to use a visual predictive model as the world model to learn the interactions between the robot and objects without making the model dependent on any specific task. The results in Table 6 show that our method still achieves a success rate of 50% on placing an object into a rectangle bowl, although in the training, the model has never seen either a demonstration of placing an object into a rectangle bowl or even the rectangle bowl itself. The intention of this experiment is not to compare an absolute winner but rather to foster an open dialog concerning whether to use inductive bias to generalize tasks or world dynamics.

Table 6 Comparison of the success rate of generalization from the task of demonstrations to a new task

	Task of demonstrations	New task
Transport network	✓(100%)	✗
Goal-conditioned transport network	✓(100%)	✗
Visual predict (ours)	60%	**50%**

Task of demonstrations: *"placing an object into a round bowl"*, the new task: *"placing an object into a rectangle bowl"*

(a) Task with round bowl (b) Task with rectangle bowl

Fig. 12 Left: we obtain 500 human demonstrations of placing an object into a round bowl. Right: the trained model performs very limitedly on the new task of "placing an object into a rectangle bowl"

Table 7 Comparison of the success rate of picking in the task with single object vs. multiple objects

	Relocate (single)	Place into a bowl (multi)
Dex-Net	**100%**	60%
Ours	90%	**80%**

Dex-Net

Dex-Net [28] is a state-of-the-art picking method that estimates the optimal picking poses from a depth image. However, it does not take into account task-related objectives, such as which object to pick up and where to place it for a specific task's goal. Nevertheless, we are interested in whether or not we can introduce Dex-Net into our approach, e.g., using it to select picking. We thus evaluate the performance of Dex-Net in the task-specific situation. We conduct this evaluation on two tasks: one is to relocate an object, with only the target object visible for the Dex-Net, and another one is to place an object into a bowl, with both the object and the bowl present in the field of view.

The results in Table 7 indicate that when only the target object is visible, Dex-Net performs well (100%) in selecting a suitable picking on this object. However, in cases where multiple objects are present in the scene, Dex-Net may not consistently identify the suitable picking on the relevant object. For example, in the task of placing an object into the bowl, Dex-Net sometimes selects a picking on the edge of the bowl, which will lead to the failure of the task. In contrast, our method can select picking on the task-relevant object more consistently (60 vs. 80%). In a nutshell, methods like Dex-Net that select the most suitable pick only by the learned geometry can be used to help select the pick in pick-and-place tasks, but they need to be well adapted to the task goal.

Conclusion and discussion

We propose a visual predictive model that learns the high-level pick-and-place actions in the real robot manipulation environment. The predictive model combines a high-level action decomposer and a video prediction network to learn the intrinsic semantic information of high-level actions. We also expand our previous work [27] and contribute a new dataset. PandaGrasp-Pick&-Place contains 5K examples of a Franka Panda robot executing pick-and-place actions. In our experiments, we find that our method outperforms a CVAE network to predict the target frame conditional on the initial frame and the pick-and-place action. By comparing different visual predictive models that are trained on high-level action or not, we find that our proposed method can substantially learn the pick-and-place actions. We then evaluate our method with sample-based optimization on several real robot tasks. Our method can find appropriate pick-and-place action, especially in the scenario where this kind of high-level action is more reasonable. We also found some limitations in our work. The introduction of the primitive of high-level actions reduces the generality compared to the vanilla Visual Foresight, resulting in a lower success rate in the task of relocating without obstacles, which may be accomplished more easily through pushing. We believe that the generalization of our method could be improved through a more general primitive or combing various primitives. Also, although learning a world dynamic is task-agnostic and more generalized, it may be more challenging than learning models for each specific task, such as the Transporter network. As the state-of-the-art video predictive and generative models [19] [21] advance, their capability to learn world dynamics will become more powerful, eventually leading to better performance in the downstream tasks.

Declarations

Conflict of interest The authors declare that we have no conflict of interest.

References

1. Babaeizadeh M, Finn C, Erhan D, Campbell R, Levine S (2018) Stochastic variational video prediction. In: 6th international conference on learning representations, ICLR 2018
2. Chen B, Wang W, Wang J (2017) Video imagination from a single image with transformation generation. In: Proceedings of the on thematic workshops of ACM multimedia 2017, pp 358–366
3. Dasari S, Ebert F, Tian S, Nair S, Bucher B, Schmeckpeper K, Singh S, Levine S, Finn C (2019) Robonet: large-scale multi-robot learning. In: CoRL
4. Dasari S, Ebert F, Tian S, Nair S, Bucher B, Schmeckpeper K, Singh S, Levine S, Finn C (2019) Robonet: large-scale multi-robot learning. CoRR arXiv:1910.11215. http://arxiv.org/abs/1910.11215
5. Deisenroth MP, Englert P, Peters J, Fox D (2014) Multi-task policy search for robotics. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3876–3881
6. Deisenroth MP, Neumann G, Peters J et al (2013) A survey on policy search for robotics. Found Trends® Robot 2(1–2):1–142
7. Denton E, Fergus R (2018) Stochastic video generation with a learned prior. In: International conference on machine learning. PMLR, pp 1174–1183
8. Divya R, Peter JD (2022) Smart healthcare system-a brain-like computing approach for analyzing the performance of detectron2 and posenet models for anomalous action detection in aged people with movement impairments. Complex Intell Syst 8(4):3021–3040
9. Ebert F, Finn C, Dasari S, Xie A, Lee A, Levine S (2018) Visual foresight: model-based deep reinforcement learning for vision-based robotic control. arXiv preprint arXiv:1812.00568
10. Ebert F, Finn C, Lee AX, Levine S (2017) Self-supervised visual planning with temporal skip connections. In: CoRL, pp 344–356
11. Finn C, Goodfellow I, Levine S (2016) Unsupervised learning for physical interaction through video prediction. In: Advances in neural information processing systems, vol 29
12. Finn C, Levine S (2017) Deep visual foresight for planning robot motion. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 2786–2793
13. Frazzoli E, Dahleh MA, Feron E (2005) Maneuver-based motion planning for nonlinear systems with symmetries. IEEE Trans Robot 21(6):1077–1091
14. Fu J, Levine S, Abbeel P (2016) One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 4019–4026
15. Gal Y, McAllister R, Rasmussen CE (2016) Improving pilco with Bayesian neural network dynamics models. In: Data-efficient machine learning workshop, vol 4. ICML, p 25
16. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the Kitti dataset. Int J Robot Res 32(11):1231–1237
17. Gu S, Holly E, Lillicrap T, Levine S (2017) Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3389–3396
18. Gualtieri M, Platt R (2018) Learning 6-DoF grasping and pick-place using attention focus. In: Conference on robot learning. PMLR, pp 477–486
19. Gupta A, Tian S, Zhang Y, Wu J, Martín-Martín R, Fei-Fei L (2022) Maskvit: masked visual pre-training for video prediction. arXiv preprint arXiv:2206.11894
20. Hafner D, Lillicrap T, Fischer I, Villegas R, Ha D, Lee H, Davidson J (2019) Learning latent dynamics for planning from pixels. In: International conference on machine learning. PMLR, pp 2555–2565
21. Ho J, Salimans T, Gritsenko A, Chan W, Norouzi M, Fleet DJ (2022) Video diffusion models. arXiv preprint arXiv:2204.03458
22. Huynh-Thu Q, Ghanbari M (2008) Scope of validity of PSNR in image/video quality assessment. Electron Lett 44(13):800–801
23. Ionescu C, Papava D, Olaru V, Sminchisescu C (2013) Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans Pattern Anal Mach Intell 36(7):1325–1339
24. Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114
25. Lee AX, Zhang R, Ebert F, Abbeel P, Finn C, Levine S (2018) Stochastic adversarial video prediction. arXiv preprint arXiv:1804.01523
26. Levine S, Koltun V (2014) Learning complex neural network policies with trajectory optimization. In: International conference on machine learning. PMLR, pp 829–837
27. Ma A, Fleytoux Y, Mouret JB, Ivaldi S (2021) VP-GO: a "light" action-conditioned visual prediction model. arXiv preprint arXiv:2109.12694
28. Mahler J, Matl M, Liu X, Li A, Gealy D, Goldberg K (2018) Dex-net 3.0: computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 5620–5627
29. Pasupa K, Kittiworapanya P, Hongngern N, Woraratpanya K (2022) Evaluation of deep learning algorithms for semantic segmentation of car parts. Complex Intell Syst 8(5):3613–3625
30. Seita D, Florence P, Tompson J, Coumans E, Sindhwani V, Goldberg K, Zeng A (2021) Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp 4568–4575
31. Sekar R, Rybkin O, Daniilidis K, Abbeel P, Hafner D, Pathak D (2020) Planning to explore via self-supervised world models. In: International conference on machine learning. PMLR, pp 8583–8592
32. Siciliano B, Khatib O (2016) Springer handbook of robotics. Springer, Berlin
33. Silver D, Hasselt H, Hessel M, Schaul T, Guez A, Harley T, Dulac-Arnold G, Reichert D, Rabinowitz N, Barreto A et al (2017) The predictron: end-to-end learning and planning. In: International conference on machine learning. PMLR, pp 3191–3199
34. Sun L, Zhao C, Yan Z, Liu P, Duckett T, Stolkin R (2018) A novel weakly-supervised approach for RGB-D-based nuclear waste object detection. IEEE Sens J 19(9):3487–3500
35. Walker J, Gupta A, Hebert M (2015) Dense optical flow prediction from a static image. In: Proceedings of the IEEE international conference on computer vision, pp 2443–2451
36. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

37. Wisspeintner T, Van Der Zant T, Iocchi L, Schiffer S (2009) Robocup@ home: scientific competition and benchmarking for domestic service robots. Interact Stud 10(3):392–426
38. Wong JM, Kee V, Le T, Wagner S, Mariottini GL, Schneider A, Hamilton L, Chipalkatty R, Hebert M, Johnson DM, et al (2017) Segicp: integrated deep semantic segmentation and pose estimation. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 5784–5789
39. Wu B, Nair S, Martin-Martin R, Fei-Fei L, Finn C (2021) Greedy hierarchical variational autoencoders for large-scale video prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2318–2328
40. Wu B, Nair S, Martin-Martin R, Fei-Fei L, Finn C (2021) Greedy hierarchical variational autoencoders for large-scale video prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2318–2328
41. Yoon Y, DeSouza GN, Kak AC (2003) Real-time tracking and pose estimation for industrial objects using geometric features. In: 2003 IEEE international conference on robotics and automation (cat. no. 03CH37422), vol 3. IEEE, pp 3473–3478
42. Zeng A, Florence P, Tompson J, Welker S, Chien J, Attarian M, Armstrong T, Krasin I, Duong D, Sindhwani V et al (2020) Transporter networks: rearranging the visual world for robotic manipulation. arXiv preprint arXiv:2010.14406
43. Zeng A, Florence P, Tompson J, Welker S, Chien J, Attarian M, Armstrong T, Krasin I, Duong D, Sindhwani V et al (2021) Transporter networks: rearranging the visual world for robotic manipulation. In: Conference on robot learning. PMLR, pp 726–747
44. Zeng A, Yu KT, Song S, Suo D, Walker E, Rodriguez A, Xiao J (2017) Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1386–1383
45. Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 586–595