Experience-Embedded Visual Foresight

Lin Yen-Chen **Maria Bauza** **Phillip Isola**
Massachusetts Institute of Technology
{yenchenl,bauza,phillipi}@mit.edu

Abstract: Visual foresight gives an agent a window into the future, which it can use to anticipate events before they happen and plan strategic behavior. Although impressive results have been achieved on video prediction in constrained settings, these models fail to generalize when confronted with unfamiliar real-world objects. In this paper, we tackle the generalization problem via fast adaptation, where we train a prediction model to quickly adapt to the observed visual dynamics of a novel object. Our method, Experience-embedded Visual Foresight (EVF), jointly learns a fast adaptation module, which encodes observed trajectories of the new object into a vector embedding, and a visual prediction model, which conditions on this embedding to generate physically plausible predictions. For evaluation, we compare our method against baselines on video prediction and benchmark its utility on two real world control tasks. We show that our method is able to quickly adapt to new visual dynamics and achieves lower error than the baselines when manipulating novel objects. Videos are available at: http://evf.csail.mit.edu/.

Keywords: Video Prediction, Model-based Reinforcement Learning, Deep Learning

1 Introduction

The ability to visualize possible futures allows an agent to consider the consequences of its actions before making costly decisions. When visual prediction is coupled with a planner, the resulting framework, visual model-predictive control (visual MPC), can perform a multitude of tasks – such as object relocation and the folding of clothes – all with the same unified model [1, 2].

Like any learned model, visual MPC works very well when predicting the dynamics of an object it has encountered many times in the past, e.g., during training. But when shown a new object, its predictions are often physically implausible and fail to be useful for planning.

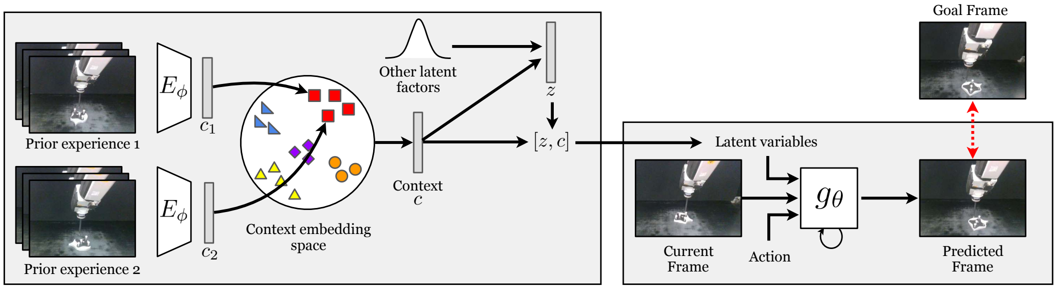
This failure contrasts with the ability of humans to quickly adapt to new scenarios. When we encounter a novel object, we may initially have little idea how it will behave – is it soft or hard, heavy or light? However, just by playing with it a bit, we can quickly estimate its basic physical properties and begin to make accurate predictions about how it will react under our control.

Our goal in this paper is to develop an algorithm that learns to quickly adapt in a similar way. We consider the problem of few-shot visual dynamics adaptation, where a visual prediction model aims to better predict the future given a few videos of previous interactions with an object.

To that end, we propose Experience-embedded Visual Foresight (EVF), where adaptation is performed via an experience encoder that takes as input a set of previous experiences with the object, in the format of videos, and outputs a low-dimensional vector embedding, which we term a "context". This embedding is then used as side information for a recurrent frame prediction module. Our method is centered on the idea that there is an embedding space of objects' properties, where similar objects (in terms of visual dynamics) are close together while different objects are far apart. Having such a space allows for few-shot visual dynamics adaptation and opens the possibility of inferring information from new and unfamiliar objects in a zero-shot fashion.

Our approach can further be formalized as a hierarchical Bayes model in which object properties are latent variables that can be inferred from a few prior experiences with the object. The full

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.



(a) Inference and sampling of latent variables

(b) Deterministic video prediction

Figure 1: **Experience-embedded Visual Foresight (EVF).** EVF allows a visual prediction model g_θ to better adapt to the visual dynamics of a novel object given a single or multiple videos of this new objects, i.e., "experiences" with it. (a) Each experience is first passed through an encoder E_ϕ to generate a compact representation c_i of the new object, where i is the index of the experience. Then, these representations are combined to create a context embedding c which is used to sample latent variable z. Both c and z are concatenated along the channel dimension to form an input to the visual prediction model g_θ. (b) The visual prediction model is a deterministic recurrent net that predicts the next frame given the current frame, the action taken, the net's recurrent state, and the latent variables inferred and sampled in (a). The experience-embedding encoder E_ϕ and video prediction model g_θ are jointly optimized.

system, as shown in Figure 1, is trained end-to-end and therefore can be understood as a form of meta-learning: under this view, the experience encoder is a fast learner, embodied in a feedforward network that modulates predictions based on just a few example videos. The weights of this fast learner are themselves also learned, but by the slower process of stochastic gradient descent over many meta-batches of examples and predictions.

EVF can be categorized as "learning from observations" [3, 4], rather than "learning from demonstrations, as it does not require that observed trajectories contain *actions*. This means it can utilize experiences when actions are unknown or generated by an agent with a different action-space.

To evaluate our approach, we compare EVF against baselines which apply gradient-based meta-learning [5] to video prediction, and validate that our method improves visual MPC's performance on two real-world control tasks in which a robot manipulates novel objects.

Our contributions include:

1. A hierarchical Bayes model for few-shot visual dynamics adaptation.
2. Improved performance on both video prediction and model-based control.
3. A learned video representation that captures properties related to object dynamics.

2 Related Work

Embedding-based Meta Learning. Learning an embedding function to summarize information from a few samples has shown promising results for few-shot learning tasks. For image classification, Matching Networks [6] learns an embedding space to compare a few labeled examples with testing data, and builds a classifier based on weighted nearest neighbor. Similarly, Prototypical Networks learns an embedding space but instead represents each class by the mean of its examples (a prototype). Then, a classifier is built using squared Euclidean distances between testing data's embedding and different classes' prototypes. For visuomotor control, TecNets [7] learns an embedding space of demo trajectories. During test time, it feeds the trajectory embeddings into a control network to perform one-shot imitation learning. We adopt a similar approach but apply it to visual MPC rather than to imitation learning. For generative modelling, Neural Statistician [8] extends the variational autoencoder (VAE) [9] to model datasets instead of individual datapoints, where the learned embedding for each dataset is used for the decoder to reconstruct samples from that dataset. Our method extends Neural Statistician [8] to model visual dynamics of diverse objects. The learned model is also used for model-based control.

Model-based Reinforcement Learning. Algorithms which learn a dynamics model to predict the future and use it to improve decision making belong to the category of model-based reinforcement learning (RL). Lately, visual model-based RL has been applied to real world robotic manipulation problems [1, 2, 10, 11]. However, these methods do not focus on adaptation of the visual prediction model. A few recent works have explored combining meta-learning with model-based RL in order to adapt to different environments in low-dimensional state space [12, 13]. In this work, we tackle the problem of adapting high-dimensional visual dynamics by learning a latent space for object properties, inspired by previous works on metric-based meta learning [6, 7].

Video Prediction. The breakthrough of deep generative models [14, 9, 15] has lead to impressive results on deterministic video prediction [16, 17, 18, 19, 20]. Specifically, action-conditioned video prediction has been shown to be useful in video games [21, 22] and robotic manipulation [1, 2, 23, 24]. Our work is most similar to approaches which perform deterministic next frame generation based on stochastic latent variables [25, 26, 27, 28]. The main difference between these works and ours is shown in Figure 2. Rather than learning a generative model for all the videos in a single dataset, we construct a hierarchical Bayes model to learn from many small yet related datasets [8, 29]. Our goal is to robustify visual MPC's performance by adapting to unseen objects.

3 Preliminaries

In visual MPC (or visual foresight), the robot first collects a dataset $D = \{\tau^{(1)}, \tau^{(2)}, ..., \tau^{(N)}\}$ which contains N trajectories. Each trajectory τ can further be represented as a sequence of images and actions $\tau = \{I_1, a_1, I_2, a_2, ..., I_T\}$. Then, the dataset is used to learn an action-conditioned video prediction model g_θ that approximates the stochastic visual dynamics $p(I_t|I_{1:t-1}, a_{t-1})$. Once the model is learned, it can be coupled with a planner to find actions that minimize an objective specified by user. In this work, we adopt the cross entropy method (CEM) as our planner and assume the objective is specified as a goal image. At the planning stage, CEM aims to find a sequence of actions that minimizes the difference between the predicted future frames and the goal image. Then, the robot executes the first action in the planned action sequence and performs replanning.

4 Method

Our goal is to robustify visual MPC's performance when manipulating unseen objects through rapid adaptation of visual dynamics. Inspired by previous works, we hypothesize that humans' capability of rapid adaptation is powered by their ability to infer latent properties that can facilitate prediction. To endow our agent the same capability, we consider a setting different from prior work where we are given K datasets D_k for $k \in [1, K]$. Each dataset $D_k = \{\tau^{(1)}, \tau^{(2)}, ..., \tau^{(N)}\}$ contains N trajectories of the robot interacting with a specific environment. We assume there is a common underlying generative process p such that the stochastic visual dynamics $p_k(I_t|I_{1:t-1}, a_{t-1})$ for dataset D_k can be written as $p_k(\cdot) = p(\cdot|c_k)$ for c_k drawn from some distribution $p(c)$, where c is a "context" embedding. Intuitively, the context c shared by all trajectories in a dataset D_i captures environmental information that generates the dataset, e.g., an object's mass and shape.

Our task can thus be split into generation and inference components. The generation component consists of learning a model g_θ that approximates the underlying generative process p. To model stochasticity, we introduce latent variables z_t at each time step to carry stochastic information about the next frame in a video. The inference component learns an approximate posterior over the context c. Since encoding the whole dataset D_i is computationally prohibitive, we construct a support set S which contains M trajectories sub-sampled from D_i and learn a function $E_\phi(c|S)$ to estimate c from S. In our experiments, M is a small number, $M \leq 5$. We denote a trajectory, image, and action in the support set as τ^S, I_t^S and a_t^S. The rightmost subfigure of Figure 2 shows the graphical model of our method. Overall, our learning approach is an extension of VAE, and we detail this formalism in Section 4.2.

4.1 Overview

Before detailing the training procedure, we explain how our method performs adaptation and generates predictions for unseen objects. To adapt the video prediction model, we first use the inference component, i.e., an experience encoder $E_\phi(c|S)$ with parameters ϕ, to encode videos in the support set S into an approximate posterior over context. Then, we sample a context from this posterior and feed it into our generation component to produce predictions. Our generation component is an

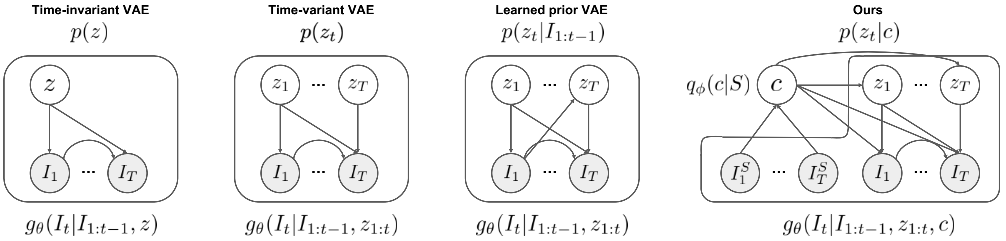


Figure 2: **Graphical models for video prediction.** Actions are omitted for ease of notation. *Left*: time-invariant model proposed in Babaeizadeh et al. [25]. *Middle Left*: time variant model adopted in previous works [25, 26, 27]. *Middle Right*: learned prior model proposed in Denton et al. [26]. *Right*: our hierarchical generative process for video prediction. Different from previous works [25, 26, 27], we introduce a latent variable c that varies between videos of different datasets but is constant for videos of the same dataset. We refer to it as the *context* which captures environmental information used to generate the dataset.

action-conditioned video prediction model g_θ parameterized by θ, that generates the next frame $\hat{I}_t$, based on previous frames $I_{1:t-1}$, action a_t, stochastic latent variables $z_{1:t}$, and context c.

The two components are trained end-to-end jointly with the aid of a separate frame encoder $q_\psi(z_t|I_{1:t}, c)$ (not used at test time). To train our model, we again first sample c from the approximate posterior constructed by the experience encoder. Then, the frame encoder encodes I_t, i.e., the target of the prediction model, along with previous frames $I_{1:t-1}$ and c to compute another distribution $q_\psi(z_t|I_{1:t}, c)$ from which we sample z_t. By assuming the prior of the context c and the latent variable z to be normally distributed, we force $E_\phi(c|S)$ and $q_\psi(z_t|I_{1:t}, c)$ to be close to the corresponding prior distribution $p(c)$ and $p(z)$ using KL-divergence terms. These two terms constrain the information that c and z_t can carry, forcing them to capture useful information for prediction. In addition, we have another term in our loss to penalize the reconstruction error between $\hat{I}_t$ and I_t.

4.2 Variational Auto-encoders Over Sets

To further explain our model, we adopt the formalism of variational auto-encoders over sets [8, 29]. For each trajecory τ, the likelihood given a context c can be written as:

$$p(\tau) = \prod_{t=1}^{T} \int g_\theta(I_t|I_{1:t-1}, a_{t-1}, z_{1:t}, c) p(z_{1:t}) \, \mathrm{d}z_{1:t} \tag{1}$$

However, this likelihood cannot be directly optimized as it involves marginalizing over the latent variables $z_{1:t}$, which is generally intractable. Therefore, the frame encoder $q_\psi(z_t|I_{1:t}, c)$ is used to approximate the posterior with a conditional Gaussian distribution $\mathcal{N}(\mu_\psi(I_{1:t}, c), \sigma_\psi(I_{1:t}, c))$. Then, the likelihood of each trajectory in a dataset can be approximated by the variational lower bound:

$$L_{\theta,\psi}(I_{1:T}) = \sum_{t=1}^{T} [\mathbb{E}_{q_\psi(z_{1:t}|I_{1:t},c)} \log g_\theta(I_t|I_{1:t-1}, a_{t-1}, z_{1:t}, c) - \beta D_{KL}(q_\psi(z_t|I_{1:t}, c)||p(z))] \tag{2}$$

Note that although Eq. (2) can be used to maximize the likelihood of trajectories in a single dataset [25, 26, 27], our goal is to approximate the commonly shared generative process p through maximizing the likelihood of many datasets. Based on the graphical model plotted on the right in Figure 2, the likelihood of a single dataset can be written as:

$$p(D) = \int p(c) \Big[\prod_{\tau \in D} \prod_{t=1}^{T} \int g_\theta(I_t|I_{1:t-1}, a_{t-1}, z_{1:t}, c) p(z_{1:t}|c) \, \mathrm{d}z_{1:t} \Big] \, \mathrm{d}c \tag{3}$$

In order to solve the intractability of marginalizing over the context c, we introduce an experience encoder $E_\phi(c|D)$ which takes as input the whole dataset to construct an approximate posterior over context. By applying variational inference, we can lower bound the likelihood of a dataset with:

$$
\begin{aligned}
L_{\theta,\phi}(D) &= \mathbb{E}_{E_\phi(c|D)}\Big[\sum_{\tau\in D}\sum_{t=1}^{T}[R_D - \beta Z_D]\Big] - \gamma C_D \ \text{ where} \\
R_D &= \mathbb{E}_{q_\psi(z_{1:t}|I_{1:t},c)} \log g_\theta(I_t|I_{1:t-1}, a_{t-1}, z_{1:t}, c) \\
Z_D &= D_{KL}(q_\psi(z_t|I_{1:t},c)||p(z)) \\
C_D &= D_{KL}(E_\phi(c|D)||p(c))
\end{aligned}
\tag{4}
$$

The hyper-parameters β and γ represent the trade-off between minimizing frame prediction error and fitting the prior. Taking smaller β and γ increases the representative power of the frame encoder and the experience encoder. Nonetheless, both may learn to simply copy the target frame I_t if β or γ are too small, resulting in low training error but struggling to generalize at test time.

4.3 Stochastic Lower Bound

Note that in Eq. (4), calculating the variational lower bound for each dataset requires passing the whole dataset through both inference and generation components for each gradient update. This can become computationally prohibitive when a dataset contains hundreds of samples. In practice, since we only pass a support set S subsampled from D, the loss is re-scaled as follows:

$$
\log p(D) \geq \mathbb{E}_{E_\phi(c|S)}\Big[\sum_{\tau\in D}\sum_{t=1}^{T}[R_D - \beta Z_D]\Big] - \gamma C_D = \sum_{\tau\in D}\Big[\mathbb{E}_{E_\phi(c|S)}\sum_{t=1}^{T}[R_D - \beta Z_D] - \frac{1}{|D|}\gamma C_D\Big]
\tag{5}
$$

In this work, we sample 5 trajectories without replacement to construct the support set S.

4.4 Model Architectures

We use SAVP [27] as our backbone architecture for the generation component. Our model g_θ is a convolutional LSTM which predicts the transformations of pixels between the current and next frame. Skip connections with the first frame are added as done in SNA [23]. Note that at every time step t, the generation component only receives I_{t-1}, a_{t-1}, z_t, and c as input. The dependencies on all previous $I_{1:t-2}$ and $z_{1:t-1}$ stem from the recurrent nature of the model. To implement the conditioning on action, latent variables, and context, we concatenate them along the channel dimension to the inputs of all the convolutional layers of the LSTM. The experience encoder is a feed-forward convolutional network which encodes the image at every time step and an LSTM which takes as input the encoded images sequentially. We apply the experience encoder to M trajectories in the support set to collect M LSTM final states. Sample mean is used as a pooling operation to collapse these states into the mean of our approximate posterior $E_\phi(c|S)$. The frame encoder is a feed-forward convolutional network which encodes I_t and I_{t-1} at every time step. We train the model using the re-parameterization trick and estimate the expectation over $q_\psi(z_{1:t}|I_{1:t}, c)$ with a single sample.

5 Experiments

In this section, we aim to answer the following questions: (i) Can we learn a context embedding c that facilitates visual dynamics adaptation? (ii) Does the context embedding c improve the performance of downstream visual MPC tasks? (iii) Can the context embedding c be used to infer similarity between objects?

First we present video prediction results on Omnipush and KTH Action to answer questions (i) and (iii) from section 5.1 to 5.2. Then, we report results on two real world control tasks to answer question (ii) in section 5.3. In our experiments, we compare our method to the following baselines:

- **SAVP.** Stochastic video prediction model trained with the VAE objective proposed in [27]. We only use VAE loss as it's more intuitive to apply finetuning or gradient-based meta learning [5] to it. Also, we found no notable performance difference.
- **SAVP-F.** SAVP finetuned with 5 videos in the support set with learning rate 5e-4 for 50 steps.
- **SAVP-MAML.** SAVP trained with MAML [5] and finetuned the same way as in SAVP-F.

We also tried implementing an RNN-based meta-learner [30], where SAVP's hidden states are not reset after encoding videos from the support set. However, it doesn't show improvement.

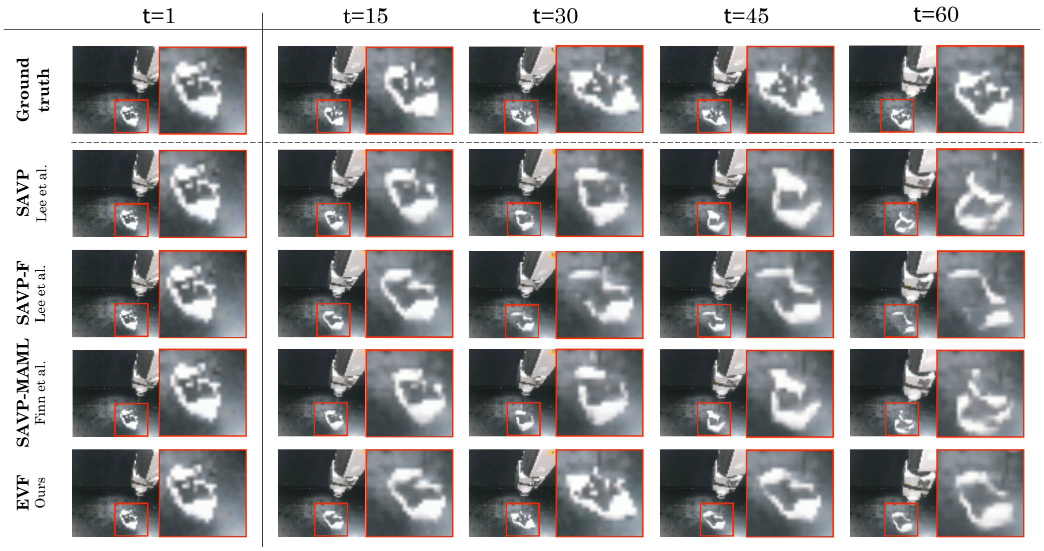


Figure 3: **Qualitative Results for Omnipush.** We show generated videos from EVF (ours), SAVP, SAVP-F, and SAVP-MAML. For clarity, patches that contain objects are cropped and shown on the right of predicted frames at each time step. Each method predicts 60 frames, with the time steps indicated at the top. *Left*: SAVP predictions deform the novel object into a circle. For both SAVP-F and SAVP-MAML, the novel object vanishes into the background. In comparison, EVF is able to predict more physically plausible futures. *Right*: the novel object completely deforms into an ellipse in all the predictions generated by baseline methods. Comparatively, EVF preserves the orientation and the shape of object better. This can be useful for visual MPC whose robustness heavily relies on the correctness of the predictions. Please refer to our supplementary material for video results.

5.1 Omnipush Dataset

Our first experiment use Omnipush dataset [31] which contains RGBD videos of an ABB IRB 120 industrial arm pushing 250 objects, each pushed 250 times. For each push, the robot selects a random direction and makes a 5cm straight push for 1s. Using modular meta-learning [32] on ground truth pose data (rather than raw RGBD), the authors showed that Omnipush is suitable for evaluating adaptation because it provides systematic variability, making it easy to study, e.g., the effects of mass distribution vs. shape on the dynamics of pushing.

We use the first split of the dataset (i.e., 70 objects without extra weight) to conduct the experiments. We condition on the first 2 frames and train to predict the next 10. To benchmark the adaptability of our approach, we train on 50 objects and test on the remaining 20 novel objects. Figure 3 shows qualitative results of video prediction. Following previous works [26, 27], we provide quantitative comparisons by computing the learned perceptual image patch similarity (LPIPS) [33], the structural similarity (SSIM) [34], and the peak signal-to-noise ratio (PSNR) between ground truth and generated video sequences. Quantitative results are shown in Figure 4. Our method predicts more physically plausible videos compared to baselines, and especially captures the shape of the pushed object better. We further show that this improves the performance of visual MPC in section 5.3.

Visualization of context embedding. In Figure 7, we use t-SNE [35] to visualize the context embedding of the 20 novel objects we tested. Despite never seeing these object during training, our model is able to produce a discriminative embedding c where objects with similar properties, such as mass and shape, are close together, and those with different properties are far apart.

5.2 KTH Action Dataset

The KTH Action dataset [36] consists of real-world videos of people performing different actions, where each person performs the same action in 4 different settings. We split the dataset into many small datasets according to person and action. Therefore, the support set consists of the other 3 videos belonging to the same small dataset. Following the setting of previous work [27], our models are trained to predict 10 future frames conditioned on 10 initial frames. During test time, it is tested

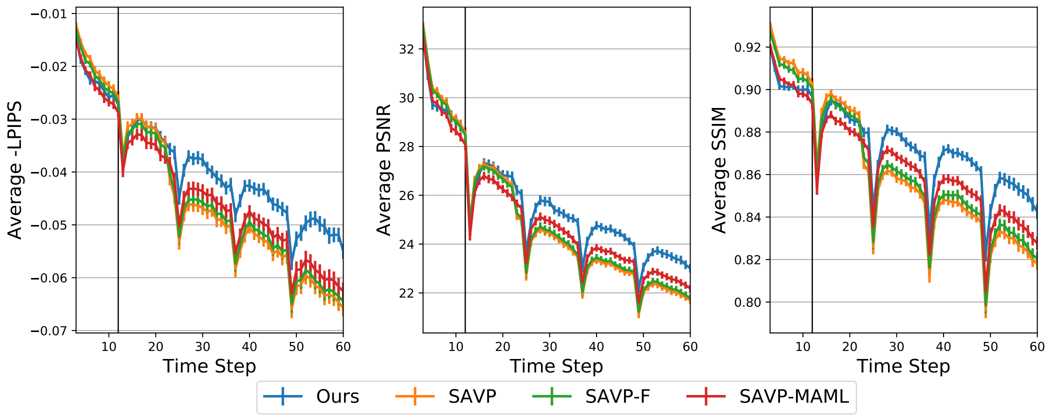


Figure 4: **Quantitative Results for Omnipush.** We show the similarity (higher is better) between ground truth and the best sample as a function of prediction time step. Three metrics – LPIPS (*left*), PSNR(*middle*), and SSIM (*right*) are used in our evaluation. Spikes appeared every 12 time steps are caused by the switch of pushing direction in Omnipush dataset.

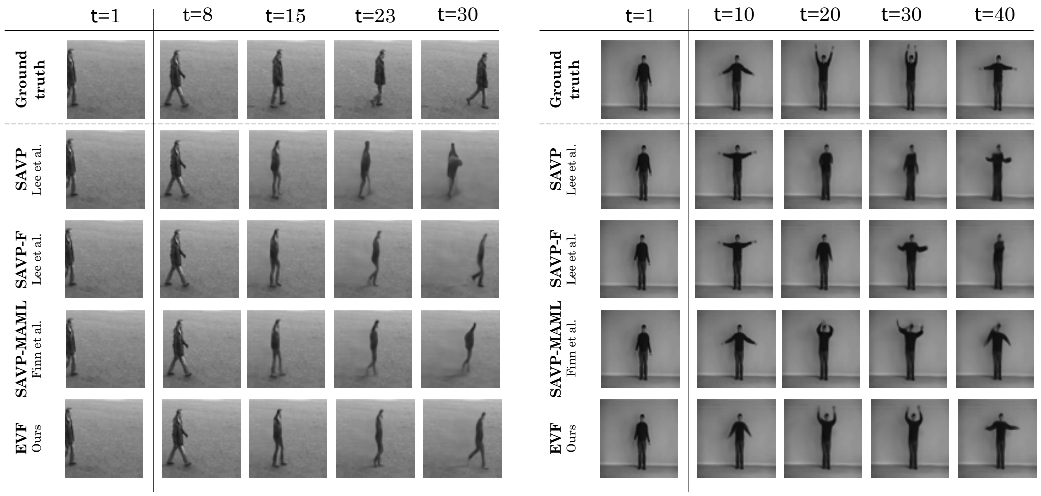


Figure 5: **Qualitative Results for KTH Action.** We show generated videos from EVF (ours), SAVP, SAVP-F, and SAVP-MAML. Each method predicts 30 frames, with the time steps indicated at the top. *Left*: SAVP's predictions deform the human body. For SAVP-MAML and SAVP-F, the human body and head are compressed in the latter time steps. In comparison, EVF is able to predict more realistic futures. *Right*: For both SAVP and SAVP-F, human's hand vanishes into its body. SAVP-MAML's predictions preserve human's hand but slightly deform its head. In comparison, EVF preserves the shape of the human better. Results on KTH show that our method can be applied to domains beyond robotics.

to predict 30 frames into the future. We show qualitative results in Figure 5 and quantitative results in Figure 6.

5.3 Real World Pushing

We perform two types of real-world robotic experiments to verify that predictions generated by our method are more physically plausible, and that this benefits visual MPC. In the first experiment, we ask the robot to perform re-location, where the robot should push an object to a designated pose. In the second experiment, the robot has to imitate the pushing trajectory presented in a 10-second video. In both tasks we use pixel-wise ℓ_2 loss, with the robot arm masked, to measure the distance between the predicted image and the goal image. At each iteration of planning, our CEM planner samples 200 candidates and refits to the best 10 samples. The results of this experiment can be found in Table 1. Note that several more advanced loss functions have been recently proposed for visual

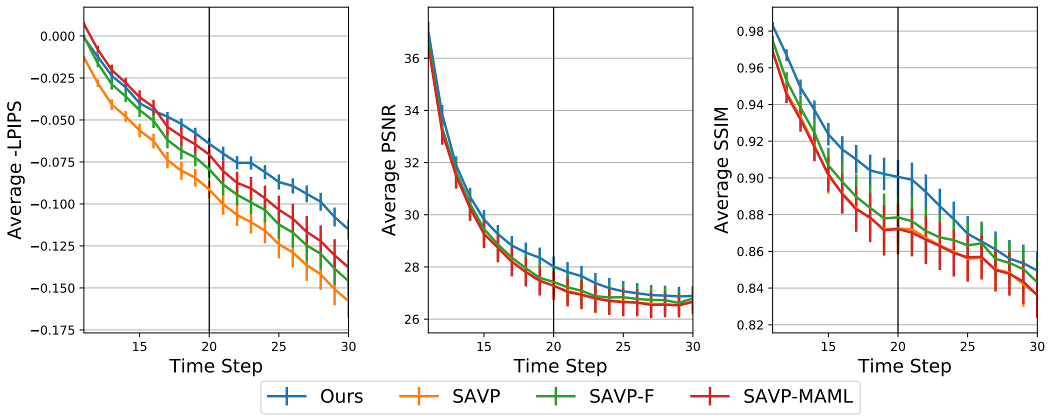


Figure 6: **Quantitative Results for KTH Action.** We show the similarity (higher is better) between ground truth and the best sample as a function of prediction time step. Three metrics – LPIPS (*left*), PSNR(*middle*), and SSIM (*right*) are used in our evaluation. .

Table 1: **Errors in re-positioning and trajectory-tracking task.** Units are millimeter. *Left*: in re-positioning, results are calculated with 5 different random seeds. *Right*: in trajectory-tracking, we report the mean error across time steps and the final offset.

Re-positioning			
Method	Type	Seen	Unseen
No motion	mean	90.4	91.9
	median	87.4	90.4
SAVP	mean	26.2	48.4
	median	27.1	48.5
EVF (ours)	mean	**21.8**	**36.2**
	median	**20.1**	**29.1**

Trajectory-tracking					
Method	Type	Traj 1	Traj 2	Traj 3	Traj 4
No motion	mean	112	93.6	91.1	97.7
	final	154	127	137	141
SAVP	mean	33.5	42.7	29.5	41.2
	final	79.8	60.8	49	59.5
EVF (ours)	mean	**31.6**	**25.2**	**21.6**	**27.5**
	final	**59.9**	**47.7**	**48.7**	**48.6**

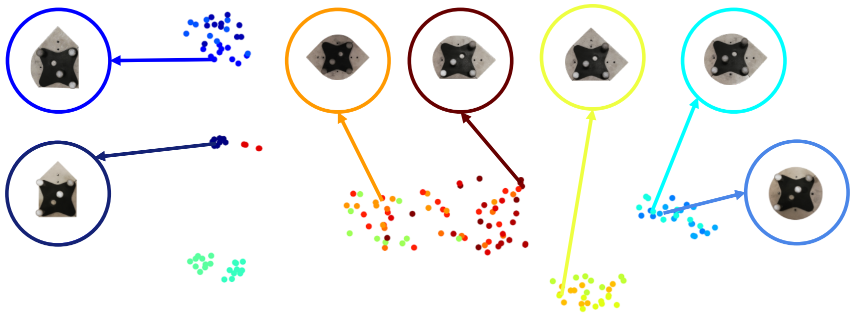


Figure 7: **t-SNE [35] results for Omnipush.** We visualize the context embedding of 20 novel objects through t-SNE. We found that embeddings are closer to each other when objects posses similar shapes and mass.

MPC [24, 23, 2], but we opted for simple ℓ_2 to isolate the contribution of our method in terms of improved visual predictions, rather than improved planning.

6 Conclusion

In this work, we considered the problem of visual dynamics adaptation in order to robustify visual MPC's performance when manipulating novel objects. We proposed a hierarchical Bayes model and showed improved results on both video prediction and model-based control. In the future, it would be interesting to investigate how to acticely collect useful data for few-shot adaptation.

Acknowledgments

We thank Alberto Rodriguez, Shuran Song, and Wei-Chiu Ma for helpful discussions. This research was supported in part by the MIT Quest for Intelligence and by iFlytek.

References

[1] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[2] 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.

[3] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *CoRR*, abs/1907.03146, 2019. URL http://arxiv.org/abs/1907.03146.

[4] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2050–2053, 2018.

[5] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[7] S. James, M. Bloesch, and A. J. Davison. Task-embedded control networks for few-shot imitation learning. *arXiv preprint arXiv:1810.03237*, 2018.

[8] H. Edwards and A. Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[9] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] A. Byravan and D. Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 173–180. IEEE, 2017.

[11] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine. Solar: deep structured representations for model-based reinforcement learning. 2018.

[12] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

[13] A. Nagabandi, C. Finn, and S. Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[16] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[17] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3560–3569. JMLR. org, 2017.

[18] N. Wichers, R. Villegas, D. Erhan, and H. Lee. Hierarchical long-term video prediction without supervision. *arXiv preprint arXiv:1806.04768*, 2018.

[19] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1771–1779. JMLR. org, 2017.

[20] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

[21] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.

[22] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[23] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017.

[24] A. Xie, A. Singh, S. Levine, and C. Finn. Few-shot goal inference for visuomotor learning and planning. In *Conference on Robot Learning*, pages 40–52, 2018.

[25] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[26] E. Denton and R. Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[27] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[28] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma. Videoflow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434*, 2019.

[29] L. B. Hewitt, M. I. Nye, A. Gane, T. Jaakkola, and J. B. Tenenbaum. The variational homoencoder: Learning to learn high capacity generative models from few examples. *arXiv preprint arXiv:1807.08919*, 2018.

[30] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rlˆ2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[31] M. Bauza, F. Alet, Y.-C. Lin, T. Lozano-Perez, L. P. Kaelbling, P. Isola, and A. Rodriguez. Omnipush: accurate, diverse, real-world dataset of pushing dynamics with rgb-d video. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019.

[32] F. Alet, T. Lozano-Pérez, and L. P. Kaelbling. Modular meta-learning. *arXiv preprint arXiv:1806.10166*, 2018.

[33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[35] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[36] C. Schldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proc. Int. Conf. Pattern Recognition (ICPR'04)*, Cambridge, U.K, 2004.