Goal-Aware Prediction: Learning to Model What Matters

Suraj Nair†, Silvio Savarese†, Chelsea Finn†
†Stanford University
Email: surajn@stanford.edu
sites.google.com/stanford.edu/gap

***Abstract*—Learned dynamics models combined with both planning and policy learning algorithms have shown promise in enabling artificial agents to learn to perform many diverse tasks with limited supervision. However, one of the fundamental challenges in using a learned forward dynamics model is the mismatch between the objective of the learned model (future state reconstruction), and that of the downstream planner or policy (completing a specified task). This issue is exacerbated by vision-based control tasks in diverse real-world environments, where the complexity of the real world dwarfs model capacity. In this paper, we propose to direct prediction towards task relevant information, enabling the model to be aware of the current task and encouraging it to only model relevant quantities of the state space, resulting in a learning objective that more closely matches the downstream task. Further, we do so in an entirely self-supervised manner, without the need for a reward function or image labels. We find that our method more effectively models the relevant parts of the scene conditioned on the goal, and as a result outperforms standard task-agnostic dynamics models and model-free reinforcement learning.**

I. Introduction

Enabling artificial agents to learn from their prior experience and generalize their knowledge to new tasks and environments remains an open and challenging problem. Unlike humans, who have the remarkable ability to quickly generalize skills to new objects and task variations, current methods in multi-task reinforcement learning require heavy supervision across many tasks before they can even begin to generalize well. One way to reduce the dependence on heavy supervision is to leverage data that the agent can collect autonomously without rewards or labels, termed *self-supervision*. One of the more promising directions in learning transferable knowledge from this unlabeled data lies in learning the dynamics of the environment, as the physics underlying the world are often consistent across scenes and tasks. However learned dynamics models do not always translate to good downstream task performance, an issue which we study and attempt to mitigate in this work.

To that end, we propose goal-aware prediction (GAP), a framework for learning forward dynamics models that direct their capacity differently conditioned on the task, resulting in a model that is more accurate on trajectories most relevant to the downstream task. Specifically, we propose to learn a latent representation of not just the state, but both the state and goal, and to learn dynamics in this latent space. Furthermore, we can learn this latent space in a way that focuses primarily on parts of the state relative to achieving the goal, namely by reconstructing the *goal-state residual* instead of the full state. We find that this modification combined with training via goal-relabeling [1] allows us to learn expressive, task-conditioned dynamics models in an *entirely self-supervised* manner. We observe that GAP learns dynamics that achieve significantly lower error on task relevant states, and as a result outperforms standard latent dynamics model learning and self-supervised model-free reinforcement learning [4] across a range of vision based control tasks.

II. Related Work

One area of past work significantly related to our work is *self-supervised reinforcement learning*, where an agent leverages data it collected autonomously to learn meaningful behaviors. Another related area is *model-based reinforcement learning*, where the agent learns a model of the dynamics of an environment, and uses it to complete a task. Lastly, like our work which studies the relationship between model error and task performance, several prior works have also explored studying *model errors* and learning better models for specific tasks. We discuss the related work in each of these areas in depth in the supplement.

III. Goal-Aware Prediction

We consider a goal-conditioned RL problem setting (described next), for which we utilize a model-based reinforcement learning approach. The key insight of this work stems from the idea that the distribution of model errors greatly affects task performance and that, when faced with limited model capacity, we can control the distribution of errors to achieve better task performance. We theoretically and empirically investigate this effect in Sections III-B and empirically in the supplement before describing our approach for skewing the distribution of model errors in Section III-C.

A. Preliminaries

We formalize our problem setting as a goal-conditioned Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, \mathcal{G}, \lambda)$ where $s \in \mathcal{S}$ is the state space, $a \in \mathcal{A}$ is the action space, $p(s_{t+1}|s_t, a_t)$ governs the environment dynamics, $p(s_0)$ corresponds to the initial state distribution, $\mathcal{G} \subset \mathcal{S}$ represents the *unknown* set of goal states which is a subset of possible states, and λ is the discount factor. Note that this is simply a special case of a Markov decision process, where we do not have access to extrinsic reward (i.e. it is self-supervised), and where we separate the state and goal for notational clarity.

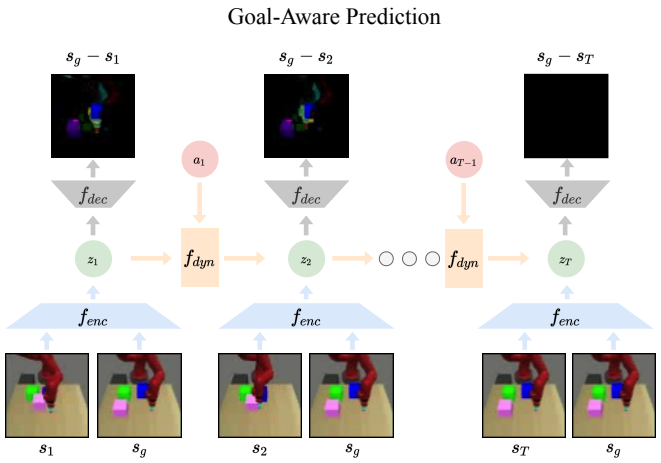

Fig. 1. **Goal-Aware Prediction:** Our proposed method, goal-aware prediction (GAP), encodes both the current state s_t and goal s_g into a single latent space z_t. Samples from the distribution of z_t are then used to reconstruct the residual between the current state and goal $s_g - s_t$. Simultaneously, we learn the forward dynamics in the latent space z, specifically, learning to predict z_{t+1} from z_t and a_t. Using this approach, we obtain 2 favorable properties: (1) the latent space only needs to capture components of the scene relevant to the goal, and (2) the prediction task becomes easier (the residual approaches 0) for states closer to the goal.

We will assume that the agent has collected an unlabeled dataset $\mathcal{D}$ that consists of N trajectories $[\tau_1, ..., \tau_N]$, and each trajectory τ consists of a sequence of state action pairs $[(s_0, a_0), (s_1, a_1), ..., (s_T)]$. We will denote the estimated distance between two states as $C(s_t, s_g) = ||s_t - s_g||_2^2$, which may not accurately reflect the true distance, e.g. when states correspond to images. At test time, the agent is initialized at a start state $s_0 \sim p(s_0)$ with a goal state s_g sampled at random from $\mathcal{G}$, and must minimize cost $\mathcal{C}(s_t, s_g)$. We assume that for any states s_t, s_g we can measure $\mathcal{C}(s_t, s_g)$ as the distance between the states, for example in image space $\mathcal{C}$ would be pixel distance. Success is measured as reaching within some true distance of s_g.

In the model-based RL setting we consider here, the agent aims to solve the RL problem by learning a model of the dynamics $p_\theta(s_{t+1}|s_t, a_t)$ from experience, and using that model to plan a sequence of actions or optimize a policy.

B. Understanding the Effect of Model Error on Task Performance

A key challenge in model-based RL is that dynamics prediction error does not directly correspond to task performance. Specifically, for good task performance, certain model errors may be more costly than others, and if errors are simply distributed uniformly over dynamics predictions, errors in these critical areas may be exploited when selecting actions downstream. Intuitively, when optimizing actions for a given task, we would like our model to to give accurate predictions for actions that are important for completing the task, while the model likely does not need to be as accurate on trajectories that are completely unrelated to the task. In this section, we formalize this intuition.

Suppose the model is used by a policy to select from N action sequences $a^i_{1:T}$, each with expected final cost $c^*_i = E_{p(s_{t+1}|s_t,a_t),a^i_{1:T}}[\mathcal{C}(s_T, s_g)]$. Without loss of generality, let $c^*_1 \leq c^*_2 ... \leq c^*_N$, i.e. the order of action sequences is sorted by their cost under the true model, which is unknown to the agent. Denote $\hat{c}_i$ as the predicted final cost of action sequence $a^i_{1:T}$ under the learned model, i.e. $\hat{c}_i = E_{p_\theta(s_{t+1}|s_t,a_t),a^i_{1:T}}[\mathcal{C}(\hat{s_T}, s_g)]$. Moreover, we consider a policy that simply selects the action sequence with lowest cost under the model: $\hat{a} = \arg\min_{a^i_{1:T}} \hat{c}_i$. *Let the policies behavior be ϵ-optimal if the cost of the selected action sequence $a^i_{1:T}$ has cost $c^*_i \leq c^*_1 + \epsilon$.* Under this set-up, we now analyze how model error affects policy performance.

Theorem III.1. *The policy will remain ϵ-optimal, that is,*

$$c^*_{i'} \leq c^*_1 + \epsilon \qquad i' = \arg\min_i \hat{c}_i \tag{1}$$

*if the following two conditions are met: first, that the model prediction error on the **best** action sequence $a^1_{1:T}$ is bounded such that*

$$|c^*_1 - \hat{c}_1| < \epsilon \tag{2}$$

and second, that the errors of sub-optimal actions sequences $a^i_{1:T}$ are bounded by

$$|c^*_i - \hat{c}_i| < (c^*_i - c^*_1) - \epsilon \quad \forall i \mid c^*_i > c^*_1 + \epsilon \tag{3}$$

Proof in supplement. Theorem III.1 suggests that, for good task performance, model error must be low for good trajectories, and we can afford higher model error for trajectories with higher cost. That is, **the greater the trajectory cost, the more model error we can afford**. Specifically, we see that the allowable error bound on cost of an action sequence from a learned model scales linearly with how far from optimal that action sequence is, in order to maintain the optimal policy for the downstream task. Note, that while Theorem III.1 relates cost prediction error (not explicitly dynamics prediction error) to planning performance, we can expect dynamics prediction error to relate to the resulting cost prediction error. We also verify this empirically in the supplement.

C. Redistributing Model Errors with Goal Aware Prediction

We propose goal-aware prediction (GAP) as a technique to re-distribute model error by learning a model that, in addition to the current state and action, s_t and a_t, is conditioned on the goal state s_g, and instead of reconstructing the future state s_{t+1}, reconstructs the difference between the future state and the goal state, that is: $p_\theta((s_g - s_{t+1})|s_t, s_g, a_t)$. Critically, to train GAP effectively, we need action sequences that are relevant to the corresponding goal. To accomplish this, we can choose to set the goal state for a given action sequence as the final state of that trajectory, i.e. using hindsight relabeling [1]. Specifically, given a trajectory $[(s_1, a_1), (s_2, a_2), ..., (s_T)]$, the goal is assigned to be the last state in the trajectory $s_g = s_T$, and for all states $\{s_t | 1 \leq t \leq T-1\}$, $p_\theta(s_t, s_g, a_t)$ is trained to reconstruct the delta to the goal $s_g - s_{t+1}$.

Our proposed GAP method has two clear benefits over standard dynamics models. First, assuming that the agent is

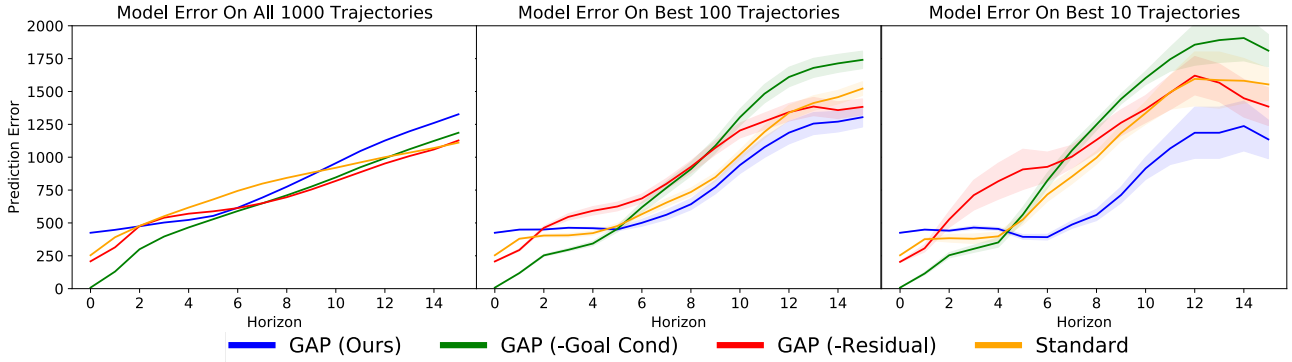

Fig. 2. **Distribution of Model Errors:** We examine the distribution of model prediction errors of GAP compared to prior methods over 1000 random action sequences, evaluated on the "Task 1" domain. The y-axis are corresponds to model mean-squared error (with standard error bars), and the x-axis corresponds to number of time steps predicted forward. Naturally, we observe that model error increases as the prediction horizon increases, for all approaches. However, although all approaches have a similar error over all 1000 action sequences (left), GAP achieves significantly lower error on the *best* 10 trajectories (right). This suggests that changing the model objective through predicting the goal-state residual leads to more accurate predictions in areas that matter in downstream tasks.

not in a highly dynamic scene with significant background distractor motion, by modeling the delta between s_g and s_t, p_θ only needs to model components of the state which are relevant to the current goal. This is particularly important in high dimensional settings where there may be large components of the state which are irrelevant to the task, and need not be modeled. Second, states s_t that are temporally close to the goal state s_g will have a smaller delta $s_g - s_t$, approaching zero along the trajectory until $s_t = s_g$. As a result, states closer to the goal will be easier to predict, biasing the model towards low error near states relevant to the goal. In light of our analysis of model error in the previous sections, we hypothesize that this model will lead to better downstream task performance compared to a standard model that distributes errors uniformly across trajectories.

IV. Experiments

In our experiments, we investigate three primary questions

(1) Does using our proposed technique for goal-aware prediction (GAP) re-distribute model error such that predictions are more accurate on good trajectories?

(2) Does re-distributing model errors using GAP result in better performance in downstream tasks?

(3) Can GAP be combined with large video prediction models to scale to the complexity of real world images?

We design our experimental set-up with these questions in mind in Section IV-A, then examine each of the questions in Sections IV-B, IV-C, and IV-D respectively.

A. Experimental Domains and Comparisons

Experimental Domains: Our primary experimental domain is a simulated tabletop manipulation task built off of the Meta-World suite of environments [5]. Specifically, it consists of a simulated Sawyer robot, and 3 blocks on a tabletop. In the *self-supervised* data collection phase, the agent executes a random policy for 2,000 episodes, collecting 100,000 frames worth of data. Then, after learning a model, the agent is tested on 4 previously unseen tasks, where the task is specified by a goal image. Details on tasks can be found in the supplement.

Comparisons: We compare our method **GAP**, to the **Standard** latent dynamics approach, an ablation **GAP (-Goal Cond)** which predicts residuals from the initial state, **GAP (-Residual)** and ablation that conditions on goals but keeps the standard objective, as well as model free **RIG** and dynamics learned via an action prediction loss termed **Inverse Model**. Further implementation details can be found in the supplement.

B. Experiment 1: Does GAP Favorably Redistribute Model Error?

In our first set of experiments, we study how GAP affects the distribution of model errors, and if it leads to lower model error on task relevant trajectories. We sample 1000 random action sequences of length 15 in the Task 1 domain. We compute the true next states $s^1_{1:H}, ..., s^{1000}_{1:H}$ and costs $c^1, ..., c^{1000}$ for each action sequence by feeding it through the true simulation environment. We then get the predicted next states from our learned models, including GAP as well the comparisons outlined above. We then examine the model error of each approach, and how it changes when looking at all trajectories, versus the lowest cost trajectories.

We present our analysis in Figure 2. We specifically look at the model error on all 1000 action sequences, the top 100 action sequences, and the top 10 action sequences. First, we observe that model error increases with the prediction horizon, which is expected due to compounding model error. More interestingly, however, we observe that while our proposed GAP approach has the highest error averaged across all 1000 action sequences, it has by far the lowest error on the top 10. This suggests that the goal conditioned prediction of the goal-state residual indeed encourages low model error in the relevant parts of the state space. Furthermore, we see that the conditioning on and reconstructing the difference to the *actual goal* is in fact critical, as the ablation GAP (-Goal Cond) which instead is conditioned on and predicts the residual to the first frame actually gets worse error on the lowest cost trajectories.

This indicates that GAP successfully re-distributes error such that it has the most accurate predictions on low-cost trajectories. We also observe this qualitatively in Figure 4. For

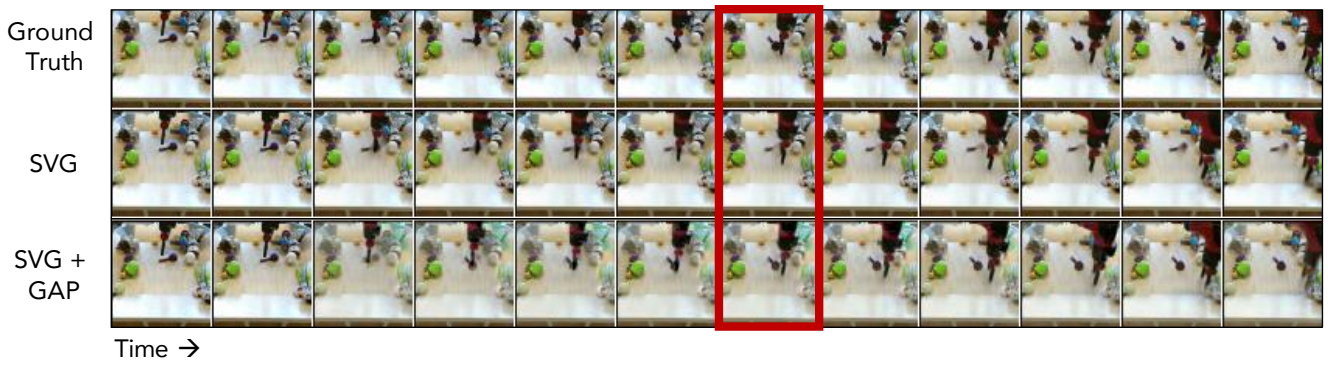

Fig. 3. **GAP+SVG Video Prediction (BAIR Robot Dataset):** Here we present qualitative examples of action-conditioned SVG with and without GAP on the BAIR robot dataset, predicting on goal-reaching trajectories. Note, in the GAP predictions the goal is added back to the predicted goal-state residual. In this case the goal is the rightmost frame. We see that GAP is able to more accurately predict the objects relevant to the goal, for example the small spoon highlighted in the red box.

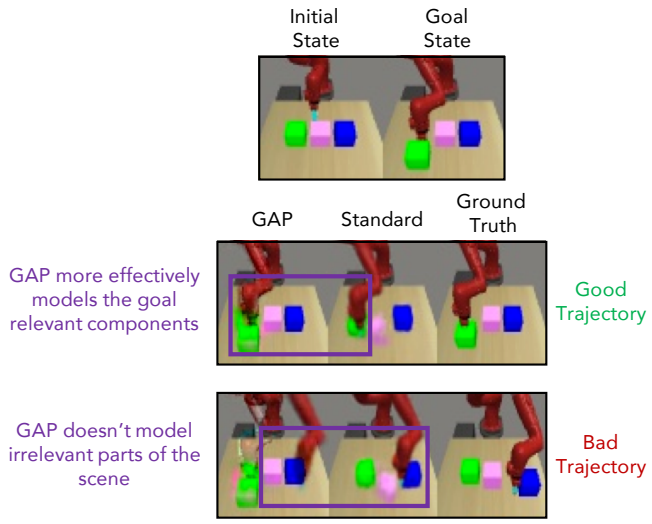

Fig. 4. **GAP Predictions on Good/Bad Trajectories.** Here we show qualitatively how GAP focuses on the task relevant parts of the scene. Note, for GAP predictions we add back the goal image to the predicted goal-state residual. Given the task specified by pushing the green block **(top)**, consider a good action sequence **(middle)** and bad action sequence **(bottom)**. On the good action sequence GAP more effectively models the goal relevant parts of the scene (the green block) than the standard model. Additionally, on the bad trajectory, GAP ignores the irrelevant objects and does not model their dynamics at all, while the standard model does.

a given initial state and goal state from Task 1, GAP effectively models the target object (the green block) on a good action sequence that reaches the goal, while the standard model struggles. On a poor action sequence that hits the non-target blocks, the Standard approach models them, while **GAP does not model interaction with these blocks at all,** suggesting that GAP does not model irrelevant parts of the scene. In the next section, we examine if this error redistribution translates to better task performance.

C. *Experiment 2: Does GAP Lead to Better Downstream Task Performance?*

To study downstream task performance, we test on the tabletop manipulation tasks described in Section IV-A. We perform planning over 30 timesteps with the learned models

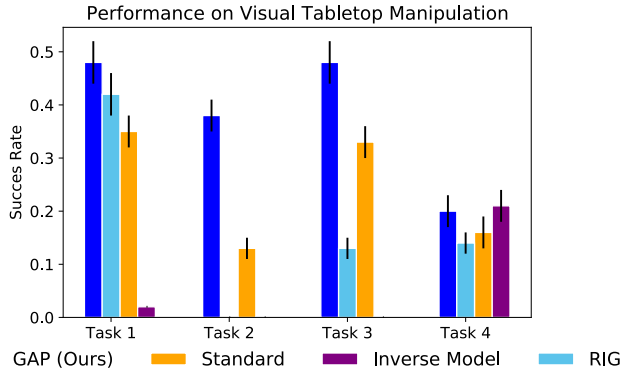

Fig. 5. **Success rate on tabletop manipulation.** On the tasks proposed in Section IV-A, we find that GAP outperforms the comparisons. Specifically on the harder 2 block manipulation task, GAP has a significantly higher success rate.

as described in Section III, and report the final success rate of each task over 200 trials in Figure 5. We see that in all tasks GAP outperforms the comparisons, especially in the most challenging 2 block manipulation task in Task 2 (where precise modeling of the relevant objects is especially important).

Hence, we can conclude that GAP not only enables lower model error in task relevant states, but by doing so, also achieves a 10-20% absolute performance improvement over all other methods on 3 out of 4 tasks.

D. *Experiment 3: Does GAP scale to real, cluttered visual scenes?*

Lastly, we study whether our proposed GAP method extends to real, cluttered visual scenes. To do so we combine it with an action-conditioned version of the video generation model, SVG [2]. Specifically, we condition the SVG encoder on the goal, and the current goal-state residual, and predict the next goal-state residual. We see that SVG+GAP is able to more effectively capture goal relevant components on the BAIR Robot Dataset [3], as shown in Figure 3, and gets lower test error on goal reaching trajectories (Supplement).

As a result, we conclude that GAP can effectively be combined with large video prediction models, and scaled to challenging real visual scenes.

ACKNOWLEDGMENTS

We would like to thank Ashwin Balakrishna, Oleh Rybkin, and members of the IRIS lab for many valuable discussions. This work was supported in part by Schmidt Futures and an NSF graduate fellowship. Chelsea Finn is a CIFAR Fellow in the Learning in Machines & Brains program.

REFERENCES

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *CoRR*, abs/1707.01495, 2017.

[2] Emily L. Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, 2018.

[3] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *CoRR*, abs/1710.05268, 2017.

[4] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pages 9191–9200, 2018.

[5] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan R Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *ArXiv*, abs/1910.10897, 2019.