AVID: Translating Human Demonstrations for Automated Learning

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, Sergey Levine
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720

Abstract

Robotic reinforcement learning (RL) holds the promise of enabling robots to learn complex behaviors through experience. However, realizing this promise requires not only effective and scalable RL algorithms but also mechanisms to reduce human burden in terms of defining the task and resetting the environment. In this paper, we study how these challenges can be alleviated with an automated robotic learning framework, where multi-stage tasks are defined simply by providing videos of a human demonstrator and then learned autonomously by the robot from raw image observations. We perform pixel-level image translation via a CycleGAN to convert the human demonstration into a video of a robot, which can then be used to construct a reward function for a model-based RL algorithm. The robot then learns the task one stage at a time, automatically learning how to reset each stage to retry it multiple times without human-provided resets. We demonstrate that our approach is capable of learning complex tasks, such as operating a coffee machine, directly from raw image observations, requiring only 20 minutes to provide human demonstrations and about 180 minutes of robot interaction with the environment. Videos of the training process and learned behavior are available from `https://sites.google.com/view/avid-neurips`.

1 Introduction

Humans and animals can learn complex tasks not just from their own experience, but also by watching other individuals. In robotics, this has been studied in the context of imitation learning [2]. However, robotic imitation learning typically utilizes demonstrations provided via the robot itself, either using teleoperation or kinesthetic teaching [4, 20, 1, 31]. In contrast, humans can learn from *watching* other people, imagining how they themselves would perform the task, and then practicing that task on their own. The question that we study is: can we endow robots with the same ability?

To this end, we present a method which we call automated visual instruction-following with demonstrations (AVID). AVID translates human videos into robot videos via CycleGAN, picks out key instruction images from the translations, and then uses image-based model-based RL to reach these instructions, automatically resetting to previous instructions each time the robot fails. This enables learning from human videos and greatly reduces the human burden for learning complex tasks in the real world. We test AVID on challenging tasks on a Sawyer robot arm, including operating a coffee machine and retrieving a cup from a cupboard. AVID outperforms ablations and prior methods in terms of data efficiency and task success, learning coffee making with only 30 human demonstrations, amounting to 20 minutes of human demonstration time, and 180 minutes of robot interaction time.

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada

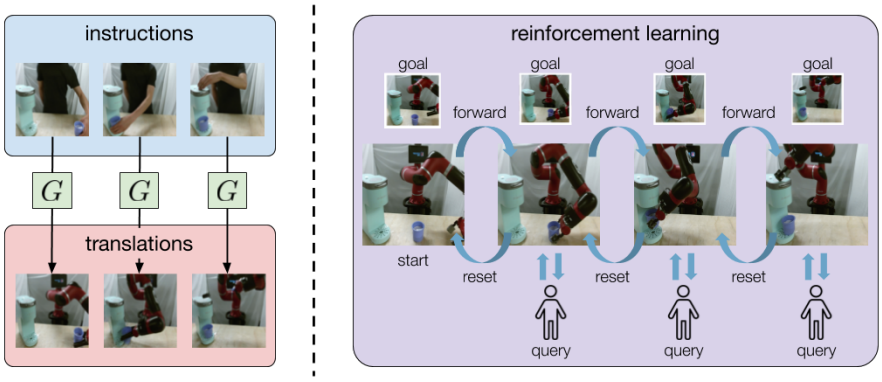

Figure 1: Schematic of the overall method. Left: Human instructions are translated at the pixel level into robot instructions via the CycleGAN. Note the artifacts in the translated images, e.g., the displaced robot gripper in the right image. Right: The robot attempts the task stage-wise via latent space planning and learned classifiers, automatically resetting and retrying until the classifier signals success, which prompts the human to confirm via a key press.

2 Automated Visual Instruction-Following with Demonstrations

In our problem setting, we assume that the task we want the robot to learn is specified through a set of human demonstration videos, where each demonstration is a trajectory of image observations depicting the human performing the task. In contrast to most prior work in robotic imitation learning, we do not assume access to robot demonstrations given through teleoperation or kinesthetic teaching. We also do not assume access to rewards provided through motion capture or other instrumented setups. Our goal is to reduce human costs in task setup by having the robot learn directly from human videos, and in this section we detail several design choices that further reduce the human burden during learning.

2.1 Translation for Goal Concept Acquisition

We provide preliminaries for CycleGAN in Appendix B. CycleGAN requires both human and robot training data, and we found that using diverse data was important for capturing a wide range of possible scenes and generating realistic translations. To that end, the training images we use from the human domain consist of both human demonstrations as well as a small amount of "random" data, where the human moves around in the scene but does not specifically attempt the task. The training images from the robot domain consist entirely of random data, as we again do not assume access to robot demonstrations. However, we collect random robot data in a few different settings, where in between settings we manually change the environment to provide more diverse data.

Once we have trained the CycleGAN model, we have a mechanism for automatically translating human demonstration videos to robot demonstration videos. However, the translated videos are imperfect and suffer from artifacts, as shown in Figure 1. As we demonstrate in Section 3, learning directly from the translated videos is susceptible to these artifacts, and as a result the final learned performance is poor. We take a different approach, where the user manually selects a set of time steps $\{t_1, \ldots, t_S\}$, and we use the images from each translated video at time step t_i to specify the i-th *instruction image*. This works well because the human demonstrations are roughly time-aligned, but we could generalize this by manually selecting corresponding images from each video, which is feasible for the modest number of demonstrations we use.

In order to specify a reward function for RL, we train classifiers for each stage $\{\mathcal{C}_1, \ldots, \mathcal{C}_S\}$, where $\mathcal{C}_i$ is provided the instruction images at time step t_i as positive examples and all other robot images, both real and translated, as negative examples. The log probabilities from the classifiers can then be used as a reward signal, similar to [10]. As mentioned earlier, providing rewards in this way requires only the translated demonstrations and does not rely on any manually designed reward shaping or instrumentation. As we discuss next, this setup lends itself naturally to a stage-wise model-based planning procedure, which makes up the core of AVID.

2.2 Model-Based RL with Instruction Images

Though we could use any RL algorithm with the classifier rewards, model-based RL has typically achieved greater data efficiency than model-free methods [7, 6]. Thus, the RL procedure that we use is the model-predictive control (MPC) baseline from SOLAR [30], which samples actions and generates rollouts using a learned model, uses the cross-entropy method (CEM) [23] to optimize the sampled actions, and executes the first action corresponding to the rollout with the highest reward, which is given by a classifier in our case. This planning is repeated at every time step to produce a trajectory of images and actions, and we encapsulate this into the `MPC-CEM` subroutine in Algorithm 1. Details for learning the model are provided in Appendix B.

When the robot is in stage s, the planner uses the log probability of $\mathcal{C}_s$ as the reward function and aims to surpass a certain classifier threshold $\mathcal{T} \in [0, 1]$, which is a hyperparameter. If this threshold is not met, the robot automatically switches to reset behavior and runs planning with $\mathcal{C}_{s-1}$, meaning that it attempts to reset to the beginning of the stage. This forward-reset behavior allows the robot to robustify its performance with very few human-provided resets, as the human only intervenes to fix problems, such as the cup falling over, rather than manually resetting each episode. The robot runs this forward-reset loop for a maximum of $\mathcal{A}$ iterations, where $\mathcal{A}$ is also a hyperparameter.

Should the threshold be met during planning, the robot will query the human user, and the user signals either success or failure through a key press. On failure, the robot switches to the reset behavior and the loop continues. On success, then the robot moves on to the next stage and repeats the same process, with $\mathcal{C}_{s+1}$ specifying the goal and $\mathcal{C}_s$ specifying the reset. This stage-wise learning avoids the compounding errors of trying to learn the entire task all at once. The full procedure is summarized in Algorithm 1 in Appendix C and concludes when the user signals success for stage S.

3 Experiments

We aim to answer the following through our experiments: (1) Is AVID capable of solving temporally extended visual tasks directly from human demonstrations? (2) What benefits, if any, do we gain from using instruction images and latent space planning? (3) What costs, if any, do we incur from not directly having access to robot demonstrations? To answer (1), we set up and evaluate on coffee making and cup retrieval tasks on a real Sawyer robot arm. To answer (2), we compare AVID to learning from full human demonstrations, using time contrastive networks (TCN) [25] and an ablation of AVID based on behavioral cloning from observations (BCO) [27], and we also compare to an ablation of AVID based on deep visual foresight (DVF) [8]. Finally, to answer (3), we evaluate BCO and behavioral cloning on the same tasks but with direct access to robot demonstrations, and we also analyze the human supervision burden of AVID. Details about comparisons can be found in Appendix D.

3.1 Experimental Setup

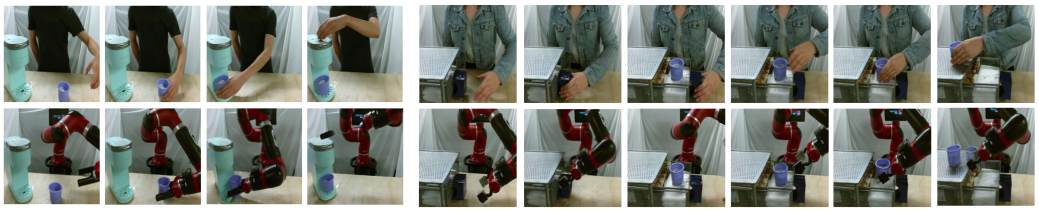

Figure 2: Sample sequence of instructions for coffee making (left) and cup retrieval (right) segmented from a human demonstration (top), translated into the robot's domain (bottom).

We evaluate our method on two temporally-extended tasks from vision: operating a personal coffee machine and retrieving a cup from a closed cupboard. These tasks illustrate that our method can learn to sequentially compose several skills by following a set of instructions extracted from a human demonstration. For all our experiments, we use end-effector velocity control on a 7 DoF Sawyer robotic manipulator, and our observations consist only of 64-by-64-by-3 RGB images. Further details about experimental setup are provided in Appendix E.

3.2 Experimental Results

Supervision	Method	Coffee making			Cup retrieval				
		Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Human Demos	AVID (ours)	**100%**	**80%**	**80%**	**100%**	**100%**	**100%**	**80%**	**70%**
	BCO ablation	70%	10%	0%	0%	0%	0%	0%	0%
	DVF ablation	60%	20%	0%	50%	50%	30%	10%	0%
	TCN [25]	10%	10%	0%	60%	20%	0%	0%	0%
Robot Demos	BCO [27]	80%	30%	0%	30%	10%	0%	0%	0%
	Behavioral Cloning	90%	90%	90%	100%	100%	60%	60%	40%

Table 1: We report the success rates up to and including each stage for both tasks, over 10 trials. The top rows are methods that learn from human demonstrations, and we bold the best performance in this category. The bottom two rows are methods trained with direct access to real robot demonstrations. AVID outperforms all other methods from human demonstrations, succeeding 8 times out of 10 on coffee making and 7 times out of 10 on cup retrieval, and even outperforms behavioral cloning from real robot demonstrations on the later stages of cup retrieval.

We report success rates for both coffee making and cup retrieval in Table 1. Success rates are evaluated using 10 trials per method per task, with no human feedback or manual intervention. Success metrics are defined consistently across all methods and are strict, e.g., placing the cup on top of the cupboard but toppling the cup over is a failure. The supplementary video shows all evaluation trials, labeled with success and failure, for each method.[1] For the final stage of coffee making, as it is difficult to visually discern whether the robot has noticeably depressed the coffee machine button, success is instead defined as being within $5\,\mathrm{cm}$ above the button, such that a pre-programmed downward motion always successfully presses the button.

For both tasks, AVID achieves the best performance among the methods that use human demonstrations, consistently learning the earlier stages perfectly and suffering less from accumulating errors in the later stages. As seen in the supplementary video, the failures of AVID for both tasks correspond to generally good behavior with small, but significant, errors, such as knocking over or narrowly missing the cup. AVID makes constant use of automated resetting and retrying during both RL and the final evaluation, and the poor performance of the DVF and BCO ablations indicates the benefits that we derive from latent space model-based RL and stage-wise training, respectively.

Despite using publicly available code from the authors and extensive hyperparameter tuning, we were unable to achieve good performance with TCN on coffee making, though TCN was moderately successful on the early stages of cup retrieval. We note that we used the single-view version of this method, and the authors also found that this version did not perform well for the tasks they experimented with [25]. The multi-view version of TCN is not applicable in our case as we do not assume multiple views of human demonstrations at training time.

In order to understand how using translated human demonstrations compares to using real robot demonstrations, we evaluate BCO and behavioral cloning from robot demonstrations obtained through teleoperation. These results are provided in Appendix F.

4 Discussion and Future Work

We presented AVID, a method for learning visual robotic multi-stage tasks directly from human demonstrations. AVID uses image-to-image translation to generate robot demonstrations and instruction images to enable a stage-wise RL procedure, which incorporates learned resets and human feedback to learn temporally extended tasks robustly and efficiently. We demonstrated that AVID is capable of learning complex tasks better than prior methods and ablations that use human demonstrations, and for the latter task AVID even outperformed behavioral cloning from real robot demonstrations. In future work, we aim to further reduce the per-task setup and burden of AVID by training a single CycleGAN on an initial large dataset, e.g., many different human and robot behaviors in a kitchen. This should enable any new task in the kitchen to be learned with just a few human demonstrations, with no additional CycleGAN training or data collection, and this is a promising direction toward truly allowing robots to learn by watching humans.

[1] `https://sites.google.com/view/avid-neurips`

References

[1] B. Akgun, M. Cakmak, J. Yoo, and A. Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *HRI*, 2012.

[2] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 2009.

[3] A. Billard and M. Mataric. Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 2001.

[4] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar. Learning collaborative manipulation tasks by demonstration using a haptic interface. In *ICAR*, 2009.

[5] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In *ICML*, 2017.

[6] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *NIPS*, 2018.

[7] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *PAMI*, 2014.

[8] 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.

[9] B. Eysenbach, S. Gu, J. Ibarz, and S. Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *ICLR*, 2018.

[10] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine. Variational inverse control with events: A general framework for data-driven reward definition. In *NIPS*, 2018.

[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2018.

[12] W. Han, S. Levine, and P. Abbeel. Learning compound multi-step controllers under unknown dynamics. In *IROS*, 2015.

[13] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal. Learning objective functions for manipulation. In *ICRA*, 2013.

[14] J. Kober and J. Peters. Learning motor primitives for robotics. In *ICRA*, 2009.

[15] A. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

[16] K. Lee, Y. Su, T. Kim, and Y. Demiris. A syntactic approach to robot imitation learning using probabilistic activity grammars. *Robotics and Autonomous Systems*, 2013.

[17] T. Lesort, N. Díaz-Rodríguez, J. Goudou, and D. Filliat. State representation learning for control: An overview. *Neural Networks*, 2018.

[18] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[19] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *ICRA*, 2018.

[20] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal. Online movement adaptation based on previous sensor experiences. In *IROS*, 2011.

[21] K. Ramirez-Amaro, M. Beetz, and G. Cheng. Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artificial Intelligence*, 2017.

[22] N. Rhinehart and K. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *ICCV*, 2017.

[23] R. Rubinstein and D. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2004.

[24] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London, Series B*, 2003.

[25] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.

[26] P. Sermanet, K. Xu, and S. Levine. Unsupervised perceptual rewards for imitation learning. In *RSS*, 2017.

[27] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. In *IJCAI*, 2018.

[28] A. Tow, N. Sünderhauf, S. Shirazi, M. Milford, and J. Leitner. What would you do? acting by learning to predict. In *IROS*, 2017.

[29] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos. Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In *AAAI*, 2015.

[30] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine. SOLAR: Deep structured representations for model-based reinforcement learning. In *ICML*, 2019.

[31] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *ICRA*, 2018.

[32] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

A Related Work

Prior methods for robotic imitation learning typically have assumed access to observations and actions that are given directly on the robot, rather than learning from human videos [3, 24, 14, 13]. To handle human videos, some prior methods have used explicit pose and object detection [16, 29, 21], predictive modeling [22, 28], context translation [19], and learning reward representations [26, 25]. Compared to these works, we explicitly handle the change in embodiment via learned pixel-level translation, and we evaluate on long-horizon multi-stage tasks. We evaluate the single-view version of time-contrastive networks (TCN) [25] on our tasks in Section 3, as this prior method also handles embodiment changes and evaluates on visual robotic tasks. TCN typically requires multiple views of human demonstrations, and we do not assume access to this in our problem setting.

Providing a reward through human videos alleviates the costs and challenges associated with real world task instrumentation, and to further reduce human burden we enable the robot to reset each stage of the task. This is similar to prior work on learning reset controllers [12, 9]. However, in contrast to this prior work, our method learns from raw image pixels and can acquire complex multi-stage tasks in the real world without manual reward design.

B Preliminaries

In order to learn from human demonstrations, our method relies on several key steps involving translating human videos, modeling robot images and actions, and extracting instruction images for model-based RL. In this section, we review the first two steps of translation and modeling, for which we borrow techniques from prior work.

B.1 Unsupervised Image-to-Image Translation

We approach human to robot translation as an unsupervised image-to-image translation problem, where the goal is to map images from a source domain X (human images, in our case) to a target domain Y (robot images) in the absence of paired training data [32, 18]. The approach we use is CycleGAN [32], which learns two mappings: $G : X \rightarrow Y$ translates source to target, and $F : Y \rightarrow X$ translates target to source. The translations are learned in order to fool discriminators D_Y and D_X which are trained to distinguish real images from translations in the target and source domains, respectively. This leads to a loss of the form

$$\mathcal{L}_{\text{GAN}}(G, D_Y) = \mathbb{E}[\log D_Y(y) + \log(1 - D_Y(G(x)))] \,,$$

And similarly for F and D_X, where x and y are source and target images drawn from the data distribution. An additional loss promotes cycle consistency between G and F, i.e.,

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}[\|x - F(G(x))\|_1 + \|y - G(F(y))\|_1] \,.$$

The overall training objective for CycleGAN is given by

$$\begin{aligned}\mathcal{L}_{\text{CG}}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y) + \mathcal{L}_{\text{GAN}}(F, D_X) \\ + \lambda \mathcal{L}_{\text{cyc}}(G, F) \,,\end{aligned}$$

Where λ is a hyperparameter. Though this approach does not exploit the temporal information that is implicit in our demonstration videos, prior work has shown that CycleGAN can successfully translate videos frame by frame, such as translating videos of horses into videos of zebras [32].

B.2 Structured Representation Learning

Prior work has shown that, for control learning in image-based domains, representation learning is an effective tool for improving data efficiency [17, 30, 11, 15]. The approach we take is to utilize a probabilistic temporally-structured latent variable model that we learn with amortized variational inference, similar to [15]. In particular, we assume the underlying state of the system $\mathbf{s}_t$ is unobserved but evolves as a function of the previous state and action, and we treat the robot images as observations $\mathbf{o}_t$ of this state. Following [15], we also decompose $\mathbf{s}_t$ into two parts, i.e., $\mathbf{s}_t = [\mathbf{s}_t^1 \; \mathbf{s}_t^2]^\top$. The generative model for $\mathbf{s}_t$ can be summarized as an initial state distribution and a learned dynamics

model, i.e.,

$$p(\mathbf{s}_1) = \mathcal{N}\left(\mathbf{s}_1^1; 0, \mathbf{I}\right) p(\mathbf{s}_1^2|\mathbf{s}_1^1)\,,$$
$$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) = p(\mathbf{s}_{t+1}^1|\mathbf{s}_t^2, \mathbf{a}_t) p(\mathbf{s}_{t+1}^2|\mathbf{s}_{t+1}^1)\,,$$

And to complete the generative model, we learn a decoder $p(\mathbf{o}_t|\mathbf{s}_t)$ which we represent as a convolutional neural network. In order to learn this model, we introduce a variational distribution $q(\mathbf{s}_{1:T}; \mathbf{o}_{1:T})$ which approximates the posterior $p(\mathbf{s}_{1:T}|\mathbf{o}_{1:T}, \mathbf{a}_{1:T})$, where the $1:T$ notation denotes entire trajectories. We use a mean field variational approximation

$$q(\mathbf{s}_{1:T}; \mathbf{o}_{1:T}) = \prod_t q(\mathbf{s}_t; \mathbf{o}_t)\,,$$

Where the encoder $q(\mathbf{s}_t; \mathbf{o}_t)$ is also a convolutional network. Similar to [15], the generative model parameters are shared with the encoder according to

$$q(\mathbf{s}_t; \mathbf{o}_t) = q(\mathbf{s}_t^1|\mathbf{o}_t) p(\mathbf{s}_t^2|\mathbf{s}_t^1)\,.$$

We jointly learn p and q via maximization of the variational lower bound (ELBO), given by

$$\begin{aligned}\text{ELBO}[p, q] = \mathbb{E}_q[p(\mathbf{o}_t|\mathbf{s}_t)] - D_{\text{KL}}(q_{\mathbf{s}_1}(\cdot; \mathbf{o}_1)\|p_{\mathbf{s}_1}) \\ - \mathbb{E}_q\left[\sum_{t=1}^{T-1} D_{\text{KL}}(q_{\mathbf{s}_{t+1}}(\cdot; \mathbf{o}_{t+1})\|p_{\mathbf{s}_{t+1}}(\cdot|\mathbf{s}_t, \mathbf{a}_t))\right]\,.\end{aligned}$$

This model provides us with an encoder and a dynamics model that we can use for encoding instruction images and for model-based planning in the learned latent space.

C Algorithm for AVID RL Forward Pass

Algorithm 1 AVID RL Forward Pass

Require: Pre-trained model $\mathcal{M}$ and $\{\mathcal{C}_i\}_{i=0}^S$, initial robot data $\mathcal{D}$
Require: Max attempts per stage $\mathcal{A}$, classifier threshold $\mathcal{T}$

`stage` $\leftarrow 1$
while `stage` $\leq S$ **do**
  `attempts` $\leftarrow 0$
  `stage-completed` $\leftarrow$ *false*
  while `attempts` $< \mathcal{A}$ **and not** `stage-completed` **do**
    $\tau \leftarrow$ `MPC-CEM`$(\mathcal{M}, \mathcal{C}_{\texttt{stage}})$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \tau$
    Increment `attempts`
    if $\mathcal{C}_{\texttt{stage}}$(last observation of τ) $> \mathcal{T}$ **then**
      if human signals success **then**
        Add last observation of τ to goal images for `stage`
        `stage-completed` $\leftarrow$ *true*
      else
        $\mathcal{D} \leftarrow \mathcal{D} \cup$ `EXPLORE`()
        Train $\mathcal{M}$ and $\{\mathcal{C}_i\}_{i=0}^S$ on $\mathcal{D}$ and goal images
      end if
      `attempts` $\leftarrow 0$
    else
      $\tau \leftarrow$ `MPC-CEM`$(\mathcal{M}, \mathcal{C}_{\texttt{stage}-1})$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \tau$
    end if
  end while
  if `stage-completed` **then**
    Increment `stage`
  else
    Train $\mathcal{M}$ and $\{\mathcal{C}_i\}_{i=0}^S$ on $\mathcal{D}$ and goal images
    `attempts` $\leftarrow 0$
  end if
end while

The full forward pass of AVID is detailed in Algorithm 1. We include several important improvements on top of this procedure in order to achieve good performance. In line 14 of Algorithm 1, if the human signals failure, the EXPLORE() subroutine is invoked, in which the robot moves randomly from the state it ended in and queries the human several times. This provides additional data within the critical area of what $\mathcal{C}_s$ believes is a success. In addition, in line 15, the current image is labeled as a negative for $\mathcal{C}_s$ and added to the dataset along with the images from random exploration. Similar to [10], further training $\mathcal{C}_s$ on this data improves its accuracy by combating the false positive problem and offsetting the artifacts in the translated images that $\mathcal{C}_s$ was initially trained with. Finally, to further automate the entire process and avoid manual human resets, after the robot completes the last stage, we run Algorithm 1 in reverse in order to get back to the initial state. We repeat this entire process until the robot can reliably complete the entire task.

D Comparisons

We compare to the following prior methods:

Time-contrastive networks [25]. We evaluate the single-view version of TCN, which learns an embedding via a temporal consistency loss in order to generalize from human demonstrations to robot execution. TCN originally used PILQR for their RL algorithm [5], though any RL algorithm can be used. For comparison purposes, we use the same RL procedure as AVID, where the only difference is that the predicted reward of model-generated trajectories is obtained by decoding these trajectories into images, using $p(\mathbf{o}_t|\mathbf{s}_t)$, and embedding these images using TCN.

BCO ablation. BCO is an imitation learning method that learns from demonstrations of only observations. BCO still assumes that the expert observations are directly from the robot, so we use the translated demonstrations for this purpose. This comparison is thus as an ablation of AVID that learns from full translated demonstrations rather than stage-wise instructions. In addition, we test BCO using real robot demonstrations, which we obtain via teleoperation.

DVF ablation. DVF is a visual model-based RL method that plans directly in pixel space rather than learning a latent space. We consider an ablation of AVID where we replace our latent space planning procedure with DVF, in order to understand if AVID achieves better performance or data efficiency as a result of this design choice.

Behavioral cloning. Finally, we consider the setting in which we have real robot demonstrations consisting of both observations and actions, where a simple but powerful imitation learning approach that we can use is behavioral cloning. This "oracle" approach has access to real demonstrations but does not utilize the interactive training procedure with learned resets and human feedback. This helps us determine what benefits AVID derives from the training procedure.

E Experimental Setup Details

We evaluate our method on two temporally-extended tasks from vision: operating a personal coffee machine and retrieving a cup from a closed cupboard. These tasks illustrate that our method can learn to sequentially compose several skills by following a set of instructions extracted from a human demonstration. For all our experiments, we use end-effector velocity control on a 7 DoF Sawyer robotic manipulator, and our observations consist only of 64-by-64-by-3 RGB images.

Coffee making. The coffee making task has three stages as depicted in Figure 2. Starting with the cup on the table, the instructions are to pick up the cup, place the cup in the machine, and press the button on the top of the machine. For this task, we used 30 human demonstrations, which corresponds to 900 images and 20 minutes of human demonstration time, along with 500 images of random human data. The CycleGAN robot data consisted of 8100 images of random robot movements, where the human changed the setting 6 times by moving the cup and placing the cup in the robot's gripper. We set the classifier threshold $\mathcal{T}$ to 0.8 and the maximum number of attempts per stage $\mathcal{A}$ to 3.

The same 8100 robot images for CycleGAN training were also used to learn the models for each method, and the BCO and DVF models received 7200 additional training images of random robot

data.[2] During RL, AVID and the BCO ablation received 3015 and 1350 training images, respectively, for online model training. TCN was given 1395 human images and 14400 robot images to learn the embedding and 3060 additional images during online training. In the setting with real robot demonstrations, BCO and behavioral cloning used 1800 and 900 total images, respectively.

Cup retrieval. Cup retrieval has five stages, also shown in Figure 2: grasping the cupboard handle, opening the cupboard, moving the arm up and out of the way, picking up the cup, and placing the cup on top of the cupboard. We found that the instruction of moving the arm was important for good performance, as the model-based planning would otherwise bump into the cupboard door. Providing this instruction was trivial with our setup, as we simply specified an additional instruction time step. For this task, we used 20 human demonstrations, amounting to 600 images and 20 minutes of human time, and 300 images of random human data. We collected 7200 images of random robot data, where the human changed the setting 11 times by moving the cup, placing the cup in the robot's gripper, and opening the cupboard. $\mathcal{T}$ and $\mathcal{A}$ were again set to 0.8 and 3.

The BCO and DVF models were given 7200 additional training images of random robot data, and during RL, AVID and the BCO ablation used 1740 and 1400 additional training images, respectively. TCN was given 900 human images and 14400 robot images to learn the embedding and 1800 additional images during online training. In the setting with real robot demonstrations, BCO used 1200 total robot images and behavioral cloning used 900 total robot images.

F Comparisons from Real Robot Demonstrations

We also ran BCO and behavioral cloning from robot demonstrations obtained through teleoperation. As expected, BCO performs better in this case than from translated demonstrations, however the performance is still significantly worse than AVID from translated demonstrations. This indicates that learning from full demonstrations, even directly obtained from the robot, suffers from accumulating errors for long-horizon tasks. We note that, to our knowledge, BCO has never been tested on tasks exhibiting multiple stages or operating directly from image observations [27]. Finally, behavioral cloning outperforms our method for coffee making but surprisingly performs worse on cup retrieval. We believe that this is because the cup retrieval task has more stages and thus is harder to learn without explicit stage-wise training. Thus, compared to AVID, behavioral cloning has more stringent assumptions and may perform worse for more complex tasks, but the drawback of AVID is that it uses an RL training phase that requires human feedback. To understand the human burden associated with this training procedure, we plot the human feedback we provide during training in Figure 3. The training phase took about an hour for each task, and in both cases AVID used less than 150 key presses to learn the task, which was easy for a human supervisor to provide. We view this as a practical way to enable robots to learn complex tasks.

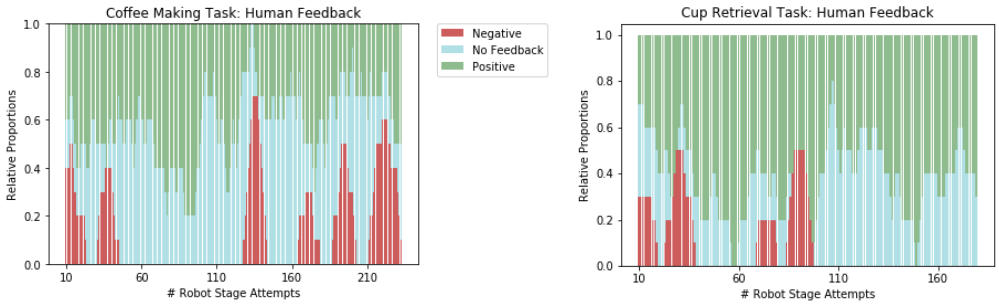

Figure 3: Visualizing human feedback during the learning process for coffee making (left) and cup retrieval (right). The x-axis is the total number of robot stage attempts, and the y-axis indicates the proportions of feedback types, smoothed over the ten most recent attempts. "No feedback" means that the classifier did not signal success and the robot automatically switches to resetting. Coffee making and cup retrieval use a total of 131 and 126 human key presses, respectively, for the learning process.

[2] We found this additional data to be important as these methods use pixel-space models which are generally less data efficient.