LEARNING AWARENESS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the setting of an agent with a fixed body interacting with an unknown and uncertain external world. We show that by maximizing the entropy of predictions about the body—touch sensors, proprioception and vestibular information—we are able to learn dynamic models of the body that can be used for control. In spite of being trained with only internally available signals, these dynamic body models come to represent external objects through the necessity of predicting their effects on the agent's own body. Our dynamics model is able to successfully predict distributions over 132 sensor readings over 100 steps into the future. We demonstrate that even when the body is no longer in contact with an object, the latent variables of the dynamics model continue to represent its shape. That is, the model learns holistic persistent representations of objects in the world, even though the only training signals are body signals. We also collect data from a real robotic hand and show that the same models can be used to answer questions about properties of objects in the real world.

1 INTRODUCTION

> *Situation awareness is the perception of the elements in the environment within a volume of time and space, and the comprehension of their meaning, and the projection of their status in the near future.* — Endsley (1987)

As artificial intelligence moves off of the server and out into the world at large; be this the simulated world, in the form of simulated walkers, climbers and other creatures (Heess et al., 2017), or the real world in the form of virtual assistants, self driving vehicles (Bojarski et al., 2016), and household robots (Jain et al., 2013); we are increasingly faced with the need to build systems that understand and reason about the world around them, and have access to this world through a bespoke interface.

When building systems like this it is natural to think of the physical world as breaking into two parts. The first part is the platform, the part we design and build, and therefore know quite a lot about; and the second part is everything else, which comprises all the strange and exciting situations that the platform might find itself in. As designers we have very little control over this part of the world, and the variety of situations that might arise are too numerous to anticipate in advance. Additionally, while the state of the platform is readily accessible (e.g. through deployment of integrated sensors), the state of the external world is generally not available to the system.

The platform hosts any sensors and actuators that are part of the system, and importantly it can be relied on to be the same across the wide variety situations where the system might be deployed. A virtual assistant can rely on having access to the camera and microphone on your smart phone, and the control system for a self driving car can assume it is controlling a specific make and model of vehicle, and that it has access to any specialized hardware installed by the manufacturer. These consistency assumptions hold regardless of what is happening in the external world.

This same partitioning of the world, into the self and the other, occurs naturally for living creatures as well. As a human being your platform is your body; it maintains a constant size and shape throughout your life (or at least these change vastly slower than the world around you), and you can hopefully rely on the fact that no matter what demands tomorrow might make of you, you will face them with the same number of fingers and toes.

The relationship between your body and the world is interesting because this interface is the critical point where information flows from the chaotic external world to the familiar internal world. Contact

with an object is where we transition from perceiving it merely through vision to also having access to its temperature, texture, mass and inertial properties through our tactile and proprioceptive senses.

This story of partitioning the world into the consistent body and the chaotic exterior parts that exchange information at the boundary suggests an approach to building models for reasoning about the world. If the body is a consistent vehicle through which an agent interacts with the world and proprioceptive and tactile senses live at the boundary of the body, then predictive models of these senses should result in models that represent external objects, in order to accurately predict their future effects on the body. Moreover, these models should be reusable in different situations because the body doesn't change.

Despite the fact that we are only modeling body signals, we show that the latent variables of the body models learn holistic persistent representations of objects in the world. Consequently, the agent is aware of the properties of objects in the world even when not touching them.

In our experiments we consider two bodies. The first is a simulated model of the hand of the Johns Hopkins Modular Prosthetic Limb (Johannes et al., 2011), realized in MuJoCo. The model is actuated by 13 motors each capable of exerting a bidirectional force on a single joint. The module is also instrumented with a series of sensors measuring angles and torques of the joints as well as pressure sensors measuring contact forces at several locations across the hand model. There are also inertial measurement units located at the end of each finger which measure translational and rotational accelerations. In total there are 132 sensor measurements whose values we predict using our dynamics model.

The second body we consider allows us to show that that our ideas apply not only in simulation, but succeed in the real world as well. Shadow Dexterous Hand, which is real robotic hand with 20 degree of freedom control. The instrumentation of the Shadow Hand is similar to the MPL model we use in simulation. It is instrumented with sensors measuring the tension of the tendons driving the fingers, and also has pressure sensors on the pad of each fingertip that measure contact forces with objects in the world. We apply the same techniques used on the simulated model to data collected from this real platform and use the resulting model to make predictions about states of external objects in the real world.

In summary, this paper shows that models trained entirely to predict body measurements can be used to answer questions about objects in the world their bodies inhabit. The information in these models can also be leveraged for control, by choosing actions to drive the world toward states where the models make uncertain predictions we gather more informative data which in turn leads to more accurate predictive models.

2 Related work

Given our goal to gather information about the world and, and in particular to actively seek out information about external objects, our work is naturally related to work on intrinsic motivation. The literature on intrinsic motivation is vast and rich, and we do not attempt to review it fully here. Some representative works include Oudeyer & Kaplan (2008; 2009); Sequeira et al. (2011); Bellemare et al. (2016) and Pathak et al. (2017).

Several authors have implemented intrinsic motivation, or curiosity based objectives in visual space, through predicting interactions with objects (Pinto et al., 2016), or through predicting summary statistics ofthe future (Venkatraman et al., 2017; Downey et al., 2017). Other authors have also investigated using learned future predictions directly for control (Dosovitskiy & Koltun, 2016).

Humans use their hands to gather information in structured task driven ways (Lederman & Klatzky, 1987); and it will become clear from the experiments why hands are relevant to our work. Our interest in hands and touch brings us into contact with a vast literature on haptics (Zheng et al., 2016; Gao et al., 2016; Cao et al., 2016; Loeb, 2013; Edmonds et al., 2017; Su et al., 2015; Navarro et al., 2012; Aggarwal et al., 2015; Liu et al.; Sung et al., 2017; Ciobanu et al., 2013; Karl et al., 2016; Su et al., 2012).

There is also work in robotics on using the anticipation of sensation to guide actions (Indranil Sur, 2017), and on showing how touch sensing can improve the performance of grasping tasks (Calan-

dra et al., 2017). Model based planning has been very successful in these domains (Deisenroth & Rasmussen, 2011).

3 MODELS AND DIAGNOSTICS

Our goal in this work is to build predictive models of proprioception and to use these models to reason about the properties of external objects. In this section we formalize this notion and discuss how we can measure whether or not we have achieved this goal.

We consider some agent operating in a discrete-time setting where there is a stochastic *unobservable* global state $s_t \in \mathcal{S}$ at each timestep t and the agent obtains a stochastic observation $o_t \in \mathcal{O}$ where $\mathcal{O} \subseteq \mathcal{S}$ and takes some action $u_t \in \mathcal{U}$. Our goal is to learn a predictive model of the agent's action-conditional future observations $p(o_{t+1:t+k}|u_{1:t+k}, o_{1:t})$ for k time steps into the future given all of the previous observations and actions it has taken. We will then use these models to reason about the global state s_t even though no information about this state is available during training.

Demonstrating success in this setting involves showing two things, first that we succeed in training accurate predictive models and second that they can be used to reason about the world. Assessing the first part is straightforward and can be done simply by computing the likelihood our models assign to a reference set of trajectories over its observations o_t.

Showing the second point requires more care, since our claim is about the information content of the states of the predictive models. We do not claim that the states of the models will be interpretable, only that the information required for reasoning about external objects is present.

To show that information required for reasoning is present in the states of our predictive models we use auxiliary models, which we call "diagnostic" models. A diagnostic model looks at the states of a proprioceptive model (which we refer to as the "base" model in this context) and uses them to to predict an interpretable quantity in the world, $x_t \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathcal{S}$ and in most cases $\mathcal{X} \cap \mathcal{O} = \emptyset$. When training a diagnostic model we allow ourselves to use privileged information x_t to define the loss, but we do not allow the diagnostic loss to influence the representations of the base model.

The diagnostic models are a post-hoc analysis strategy. The base models are trained using only the proprioceptive information, and then frozen. After the base models are trained we train diagnostic models on their states, and the claim is that if we can successfully predict properties of external objects using diagnostic models trained in this way then information about the external objects is available in the states of the base model.

We consider two types of diagnostic model in this work. The first is a property-based diagnostic where the task is to predict a physical property of some external object (e.g. its shape or spatial extent). In order to define the objective for these diagnostic tasks in simulation we can read the desired ground truth directly from the simulator state. To build a property based diagnostic in the real world we build an experimental apparatus that records the desired diagnostic quantity in addition to the proprioceptive measurements.

4 AWARENESS MODELS

We treat predicting proprioception as a sequence-to-sequence problem, where the input sequence is a prefix of an episode and the encoder observes both the proprioceptive observations o_t and the actions u_t taken at each timestep. Probabilistic graphic modeling provides many ways to model the observation space $p(o_{t+1:t+k}|u_{1:t+k}, o_{1:t})$. Our approach shown in Figure 1 that allows us to easily obtain global state knowledge is to use deterministic hidden states and divide the model into an encoder part that encodes the past into some hidden state h_t and can decode into the future with hidden states z_t.

The encoder is trained to encode the past trajectory $\{u_{1:t}, o_{1:t}\}$ into h_t. It does this by recurrently updating itself based on the current observation and action: $h_{t+1} = \text{Encoder}_\theta(o_t, u_t, h_t)$. The decoder then receives this updated hidden state.

The decoder in our model is also recurrent and is conditioned on future actions, but it is intentionally not autoregressive in time: $z_t = \text{Decoder}_\theta(u_t, z_{t-1})$. Conditioned on its hidden state, predictions

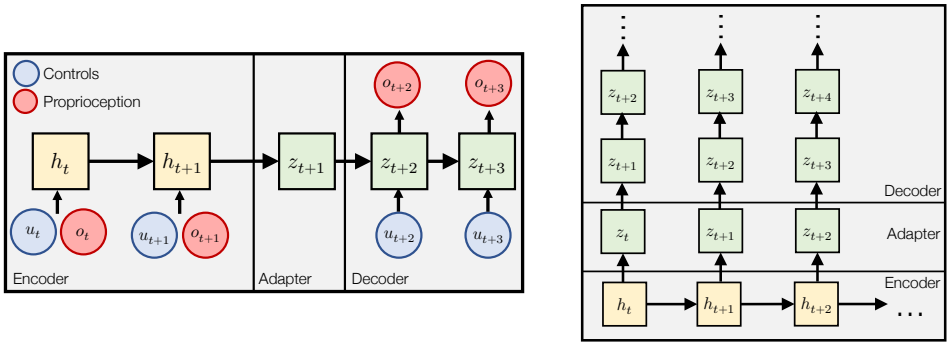

Figure 1: **Left:** Diagram of the full model. **Right:** Top down view of the training graph showing only the cores and adapters at the interface between the encoder and decoder components.

made by the decoder at each future timestep are independent, meaning the model makes open loop predictions. The decoder is trained to predict the continuation of the proprioceptive signal for k steps into the future, conditioned on the actions that were taken,

$$p(o_{t+1:t+k}|u_{t+1:t+k}, z_{t+1:t+k}) = \prod_{\kappa=1}^{k} p(o_{t+\kappa}|u_{t+\kappa}, z_{t+1:t+\kappa}) \ .$$

Between the encoder and decoder we have an adapter module that maps the final state of the encoder h_t to an initial state for the decoder z_{t+1}.

For training, there is no natural point within an episode to switch from encoding to prediction, and we use this opportunity to make forward predictions from *every* point, using a shared encoder to condition forward predictions at every step. The structure of the resulting graph is shown on the right of Figure 1. We call this technique *overshooting*, and we call the number of steps predicted forward by the decoder the overshooting length.

The effect of making open-loop (rather than closed-loop) predictions is two fold. First, it forces decisions about possible states of the external world to be made in the encoder rather than the decoder. An autoregressive model could make these decisions based on sampled future observations, but without the autoregressive connections the predicted trajectory is fixed given the encoder state and the future actions.

Second, forcing open loop predictions ensures that any true uncertainty in the state of the external world will manifest as high entropy in the decoder likelihoods. An autoregressive model can resolve this type of uncertainty through sampling, but our model is forced to make predictions that cover all possibilities. We show later sections that we can use high entropy in the predictions as an objective for control that allows us to plan actions to gather information about the environment.

All the recurrent parts of the models use single layer LSTM cores. All observations are embedded with an observation-type-specific encoder MLP (e.g. controls and sensors are each embedded with their own network in each part of the model where they are consumed). When a module takes more than one input (e.g. the encoder consumes both controls and proprioception) we embed each signal separately and concatenate the embeddings.

For making predictions we use independent mixtures of Gaussians at every step. Each dimension of each prediction is an independent mixture. We use separate MLPs to produce the the means, standard deviations and mixture weights from the output of the decoder LSTM.

We run a parameter search over several model hyperparameters for each experiment separately. The particular choices of parameters we use for each experiment are listed in Appendix C.

5 Environment

The environment consists of a simulated model of the hand part of the Johns Hopkins Modular Prosthetic Limb (Johannes et al., 2011) which we refer to as the "MPL hand", or simply "the hand".

This model is distributed with the MuJoCo HAPTIX software and is available for download from the MuJoCo website.[1] The hand is actuated by 13 motors and has sensors that provide a 132 dimensional observation. See Appendix B for a detailed description.

In each episode the hand starts suspended above the table with its palm facing downwards. A random geometric object is placed underneath the hand, which is free to move to grasp or manipulate the object. The shapes of the object is randomly chosen to be a box, cylinder or ellipsoid and the size and orientation of each object are randomly chosen from reasonable ranges.

6 PASSIVE AWARENESS

We begin by exploring awareness in the passive setting, as a pure supervised learning problem. In this setting we still build action-dependent models, but where actions are needed they are generated from a pre-programmed behavior that follows a sensible trajectory for the environment (we consider actively controlling actions in the next section). We hand-design a simple grasping motion that closes the hand about the object beneath it and then releases it. We generate data from the environment by running this hand designed grasp-and-release cycle three times for each episode.

We train one of the models described in Section 4. The full set of hyperparameters can be found in Appendix C.

We show that information about the target object can be recovered from the encoder's hidden states h_t using diagnostic model approach described in Section 3. We look at two different diagnostic tasks:

1. **Classification:** Predicting the class of the object. Each object is either a box, a cylinder or an ellipsoid.
2. **Regression:** Predicting the spatial extents of the target object. We use a separate diagnostic network for predicting the extents of each type of object. During training this is implemented by training each diagnostic model on only episodes containing the corresponding object type, and at test time this is implemented by first applying the classification diagnostic to identify the object type and then applying the appropriate shape diagnostic.

Results of this experiment are shown in Figure 2. The results show that we can reliably recover the object type and parameters from the learned dynamics models, even though this information is not available to the model directly either in the input or in the training loss.

7 ACTIVE AWARENESS

In the previous section we showed that our models come to represent properties of the external world based solely on modelling the dynamics of proprioception. By training the models to make open loop predictions about future trajectories of proprioception we arrive at models that encode information about objects in the external world that they do not directly observe. We showed that this is the case by training diagnostic models, which look at the trained states and accurately predict properties of the external objects.

In this section we go a step further and show that not only do the models represent information about the external world, but this information is accessible, in the sense that we can use the models to reason about external objects without explicitly knowing what they are. In particular, we show how the models from the previous section can be used to choose actions to gather information about the external objects. We show that this active exploration leads to better predictive models and causes them to form more accurate representations of the external objects.

[1] http://www.mujoco.org/book/haptix.html

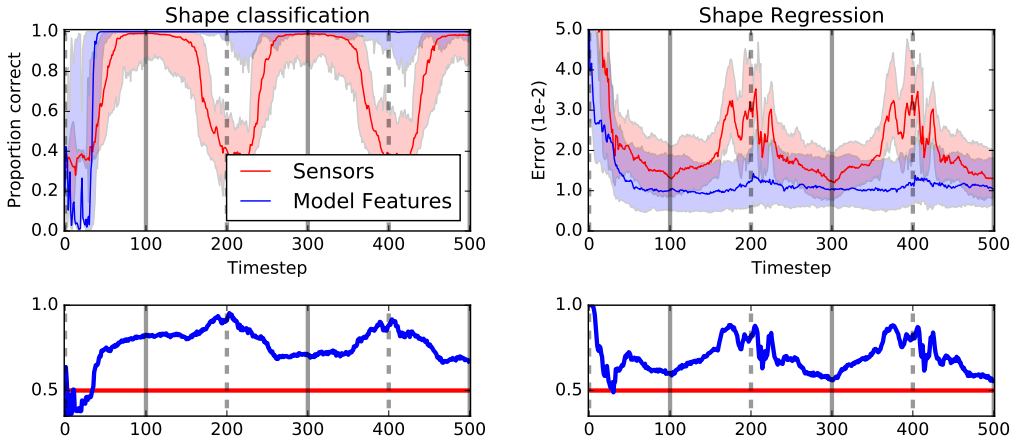

Figure 2: **Top:** Passive diagnostic curves on the Blocks dataset. The left plot shows the performance of the classification diagnostic and the right plot shows performance of the regression diagnostic. In both cases predictions based on model features (in blue) are compared with performance of the same diagnostic looking only at the sensor observations. Solid lines show the median error at each timestep, averaged over trajectoies from the test set. The shaded regions enclose the 25th-75th percentile predictions. Black lines mark critical points in the trajectories: dashed lines show when the hand is fully open and solid lines show where it is fully closed. **Bottom:** Bootstrap estimates of the probability that the model features give a better estimate of the the corresponding diagnostic quantity than the sensor readings alone, again as a function of time.

7.1 Using awareness models for control

We use Model Predictive Control (MPC) to achieve objecties with our awereness models. We express objectives as a cost function $C(z_t, f_t)$ where z_t is the decoder's hidden state at some timestep t and f_t is the predicted PDF over the observations.[2] Once the agent has taken some action u_t, obtained an observaiton o_t, and updated the encoder's hidden state h_t, the next action to take is found by solving the MPC problem

$$\begin{aligned} u^*_{1:T} \;=\; \underset{u_{1:T}, z_{1:T}, f_{1:T}}{\operatorname{argmin}} \quad & \sum_t C(z_t, f_t) \\ \text{s.t.} \quad & f_t = \text{DecodePDF}_\theta(z_t) \\ & z_t = \text{Decoder}_\theta(u_t, z_{t-1}) \\ & z_0 = \text{Adaptor}_\theta(h_t) \\ & u_{1:T} \in \mathcal{U}_{1:T} \end{aligned} \tag{1}$$

where the nominal timesteps are normalized so the first predicted timestep is at index 1 and $\text{DecodePDF}_\theta(z_t)$ returns a PDF over o_t. In most of our experiments, we parameterize DecodePDF_θ as a mixture of Gaussian distributions, ensure actions lie in a box $||u_t||_\infty \leq 1$, and enforce slew rate constraints $||u_{t+1} - u_t||_\infty \leq 0.1$.

7.2 Gathering information through maximizing uncertainty

The only source of randomness in our environment is over the properties of the target objects. The hand is always initialized in the same position and the physics simulation is deterministic. This means when our models make uncertain predictions, the source of that uncertainty comes from one of two places: (1) the model is poor, which comes either from too little data or from too small capacity, or (2) some property of the external objects are not yet resolved by the observations seen so far.

[2] We have included f_t here for notational convenience even though it is often a deterministic function of z_t.

We can exploit this fact by choosing actions to maximize the uncertainty in the rollout predictions. An agent using this uncertainty maximizing policy attempts to seek actions for which the outcome is not yet known. This uncertainty can is then resolved by executing these actions and observing their outcome, and the resulting trajectory of observations, actions, and sensations can be used to refine the model.

To choose actions to gather information we use Model Predictive Control (MPC) over an objective that maximizes the uncertainty in the predictions. We use the Rényi entropy of our model predictions as our measure of uncertainty, since it can be easily computed in closed form for the Mixture of Gaussian predictions that we make for each sensor. Concretely, for a single Mixture of Gaussians prediction $f(x)$ we can write

$$H(f) = -\log\left[\int f(x)^2 \,\mathrm{d}x\right] = -\log\left[\sum_{ij} \alpha_i \alpha_j \frac{\exp\left\{-\frac{(\mu_i-\mu_j)^2}{2\left(\sigma_i^2+\sigma_j^2\right)}\right\}}{\sqrt{2\pi}\sqrt{\sigma_i^2+\sigma_j^2}}\right]$$

where i and j index the mixture components in the likelihood. We obtain an information seeking objective by summing the entropy of the predictions across sensors and across time.

We implement this information gathering policy to collect training data for the model in which it is planning. In our implementation these are two processes running in parallel: we have several actors each with a copy of the current model weights. These use MPC as described in Section 7.1 to plan and execute a trajectory of actions that maximizes the model's predicted uncertainty over a fixed horizon trajectory into the future. The observations and actions generated by the actors are collected into a large shared buffer and stored for the learner.

While the actors are collecting data, a single learner process samples batches of the collected trajectories from the buffer being written to by the actors. The learner trains the model by maximum likelihood, using the same objective as in Section 6, and the updated model propagates back to the actors who continue to plan using the updated model (Anonymous ICLR Submission, 2017).

We train a large number of models in this way, since each one tends to generate qualitatively different behavior, even when optimizing the same information seeking objective. Different behavior between models is a result of different random initialization leading them to each collect and train on different data, and therefore each model makes different errors.

After training many models in this way, we take all of the trained models and generate new trajectories from all of the models using the same planning objective we used for training. We collect these new trajectories into a new data set that contains trajectories generated from many different models across many different instances of the environment. This data set contains substantially more diversity in behavior than the data from fixed grasps that we used in Section 6.

7.3 Achieving new objectives

We can use the trained models from the previous section to execute new behaviors, provided that they are expressible in terms of predictions made by the model. We do this with MPC as described in Section 7.1.

1. **Maximizing entropy of the predictions,** as we did during training, leads to exploratory behavior. In Figure 3 we show a typical frame from an entropy maximizing trajectory, as well as typical frames from controlling for two different objectives.
2. **Optimizing for fingertip pressure** tends to lead to grasping behavior, since the easiest way to achieve pressure on the fingertips is to push them against the target block. There is an alternative solution which is often found where the hand makes a tight fist, pushing its fingertips into its own palm.
3. **Controlling to minimize the prediction entropy** is also quite interesting. This is the negation of the information gathering objective, and it attempts to make future observations as uninformative as possible. Optimizing for this objective results in behavior where the hand consistently pulls away from the target object.

(a) Maximize predicted entropy (b) Maximize fingertip pressure (c) Minimize predicted entropy

Figure 3: Acting to optimize different objectives at test time leads to different behaviors.

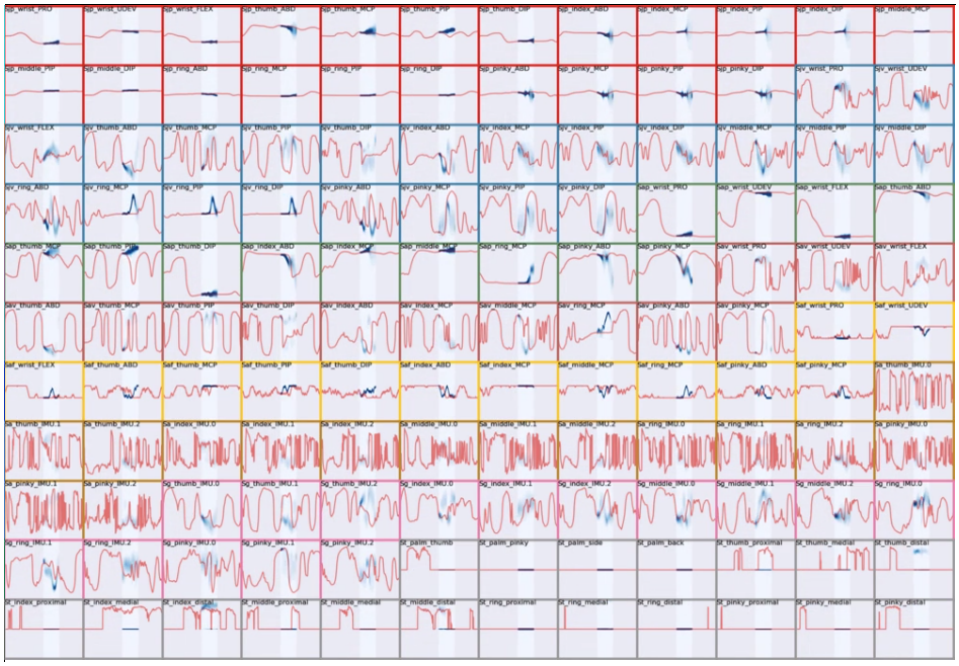

Figure 4: A visualization of the model planning to maximize predicted entropy. Each plot shows one of the 132 sensors. The red line in each plot shows the actual sensor readings achieved in this 500 step episode. The blue shaded regions show the predicted distributions unrolled over a 100 step horizon.

Figure 4 shows a snapshot of the model's internal state during planning. Each panel shows the trajectory of a different sensor reading across a 500 step episode. The figure also shows the predicted sensor readings, unrolled for 100 steps from a point mid-episode. These predictions often do not match the red lines because they show predictions along the current planned trajectory, and the plan is iteratively refined at each timestep.

8 Passive awareness in the real world on the shadow hand

We have shown that our models work well in simulation. We now turn to demonstrating that they are effective in reality as well.

We use the 24-joint Shadow Dexterous Hand[3] with 20-DOF tendon position control and set up a real life analog of our simulated environment, as shown in Figure 5. Since varying the spatial extents of an object in real life would be very labor intensive we instead use a single object fixed to a turntable

[3] https://www.shadowrobot.com/products/dexterous-hand/

that can rotate to any one of 255 orientations, and our diagnostic task in this environment is to recover the orientation of the grasped object.

We built a turntable mechanism for orienting the object beneath the hand, and design some randomized grasp trajectories for the hand to close around the block. The object is a soft foam wedge (the shape is chosen to have an unambiguous orientation) and fixed to the turntable. At each episode we turn the table to a randomly chosen orientation and execute two grasp release cycles with the hand robot.

Over the course of two days we collected 1140 grasp trajectories in three sessions of 47, 393 and 700 trajectories. We use the 47 trajectories from the initial session as test data, and use the remaining 1093 trajectories for training. Each trajectory is 81 frames long and consists of two grasp-release cycles with the target object at a fixed orientation. At each timestep we measure four different proprioceptive features from the robot:

1. The **actions** which is a set of 20 desired joint positions sent to the robot for the current timestep.
2. The **angles** which is a set of 24 measured joint positions reported by the robot at the current timestep. There are more angles than actions because not all joints of the hand are separately actuated, and the measured angles may not match the intended actions due to force limits imposed by the low level controller.
3. The **efforts**, which provide 20 distinct torque readings. Each effort measurement is the signed difference in tension between tendons on the inside and outside of one of the actuated joints.
4. The **pressures** are five scalar measurements that indicate the pressure experienced by the pads on the end of each finger.

Joint ranges of the hand are limited to prevent fingers pushing each other, and the actuator strengths are limited for the safety of the robot and the apparatus. At each grasp-release cycle final grasped and released positions are sampled from handcrafted distributions. Position targets sent to the robot are calculated by interpolating between these two positions in 20 steps.

There are multiple complexities the sensor model needs to deal with. First of all once a finger touches the object actual positions and target positions do not match, and the foam object bends and deforms. Also the hand can occasionally overcome the resistance in the turntable motor causing the target object to rotate during the episode (for about 10-20 degrees and rarely more). This creates extra unrecorded source of error in the data.

We train a forward model on the collected data, and then treat prediction of the orientation of the block as a diagnostic task. Figure 5 shows that we can successfully predict the orientation of the block from the dynamics model state.

9 Conclusion

In this paper we showed that learning a forward predictive model of proprioception we obtain models that can be used to answer questions and reason about objects in the external world. We demonstrated this in simulation with a series of diagnostic tasks where we use the model features to identify properties of external objects, and also with a control task where we show that we can plan in the model to achieve objectives that were not seen during training.

We also showed that the same principles we applied to our simulated models are also successful in reality. We collected data from a real robotic platform and used the same modelling techniques to predict the orientation of a grasped block.

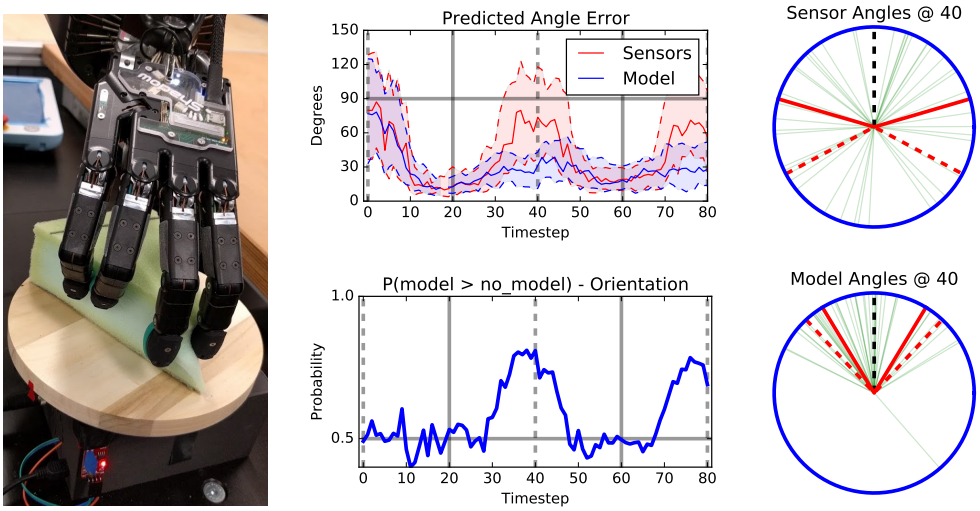

Figure 5: **Left:** The robotic hand setup. **Center:** Results on predicting block orientation with sensor data recorded from the shadow hand. The upper plot shows the median error as a function of time and the bottom plot shows a bootstrap estimate of the probability that using the model features improves on using sensor measurements directly. **Right:** Predicted angles on test trajectories at step 40 using only sensor readings (top) and model features (bottom). Green lines show predicted angles for individual samples (rotated so ground truth is vertical). The solid and dashed red lines show 50 and 75 percentile error cones, respectively.

REFERENCES

Achint Aggarwal, Peter Kampmann, Johannes Lemburg, and Frank Kirchner. Haptic object recognition in underwater and deep-sea environments. *Journal of field robotics*, 32(1):167–185, 2015.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H. Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017.

Lele Cao, Ramamohanarao Kotagiri, Fuchun Sun, Hongbo Li, Wenbing Huang, and Zay Maung Maung Aye. Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3337–3345. AAAI Press, 2016.

Vlad Ciobanu, Adrian Petrescu, Norman Hendrich, and Jianwei Zhang. Tactile sensor value preprocessing pipeline. In *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, pp. 674–680. IEEE, 2013.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. *arXiv preprint arXiv:1611.01779*, 2016.

Carlton Downey, Ahmed Hefny, Boyue Li, Byron Boots, and Geoffrey Gordon. Predictive state recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In *International Conference on Intelligent Robots and Systems (IROS), IEEE*, 2017.

Mica R Endsley. Sagat: A methodology for the measurement of situation awareness (nor doc 87-83). *Hawthorne, CA: Northrop Corporation*, 1987.

Yang Gao, Lisa Anne Hendricks, Katherine J Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 536–543. IEEE, 2016.

Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Heni Ben Amor Indranil Sur. Robots that anticipate pain: Anticipating physical perturbations from visual cues through deep predictive models. In *IROS*, 2017.

Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in neural information processing systems*, pp. 575–583, 2013.

Matthew S Johannes, John D Bigelow, James M Burck, Stuart D Harshbarger, Matthew V Kozlowski, and Thomas Van Doren. An overview of the developmental process for the modular prosthetic limb. *Johns Hopkins APL Technical Digest*, 30(3):207–216, 2011.

Maximilian Karl, Justin Bayer, and Patrick van der Smagt. Unsupervised preprocessing for tactile data. *arXiv preprint arXiv:1606.07312*, 2016.

Susan J Lederman and Roberta L Klatzky. Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368, 1987.

Chang Liu, Fuchun Sun, and Alan Yuille. Haptic object recognition: A recurrent approach.

Gerald E Loeb. Estimating point of contact, force and torque in a biomimetic tactile sensor with deformable skin. 2013.

Stefan Escaida Navarro, Nicolas Gorges, Heinz Wörn, Julian Schill, Tamim Asfour, and Rüdiger Dillmann. Haptic object recognition for multi-fingered robot hands. In *Haptics Symposium (HAPTICS), 2012 IEEE*, pp. 497–502. IEEE, 2012.

Pierre-Yves Oudeyer and Frederic Kaplan. How can we define intrinsic motivation? In *Proceedings of the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Lund University Cognitive Studies, Lund: LUCS, Brighton*. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML) 2017*, 2017.

Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *European Conference on Computer Vision*, pp. 3–18. Springer, 2016.

Pedro Sequeira, Francisco S Melo, and Ana Paiva. Emotion-based intrinsic motivation for reinforcement learning agents. In *International Conference on Affective Computing and Intelligent Interaction*, pp. 326–336. Springer, 2011.

Zhe Su, Jeremy A Fishel, Tomonori Yamamoto, and Gerald E Loeb. Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in neurorobotics*, 6, 2012.

Zhe Su, Karol Hausman, Yevgen Chebotar, Artem Molchanov, Gerald E Loeb, Gaurav S Sukhatme, and Stefan Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 297–303. IEEE, 2015.

Jaeyong Sung, J Kenneth Salisbury, and Ashutosh Saxena. Learning to represent haptic feedback for partially-observable tasks. *arXiv preprint arXiv:1705.06243*, 2017.

Arun Venkatraman, Nicholas Rhinehart, Wen Sun, Lerrel Pinto, Martial Hebert, Byron Boots, Kris M Kitani, and J Andrew Bagnell. Predictive-state decoders: Encoding the future into recurrent networks. In *Advances in Neural Information Processing Systems*, 2017.

Haitian Zheng, Lu Fang, Mengqi Ji, Matti Strese, Yigitcan Özer, and Eckehard Steinbach. Deep learning for surface material classification using haptic and visual information. *IEEE Transactions on Multimedia*, 18 (12):2407–2416, 2016.

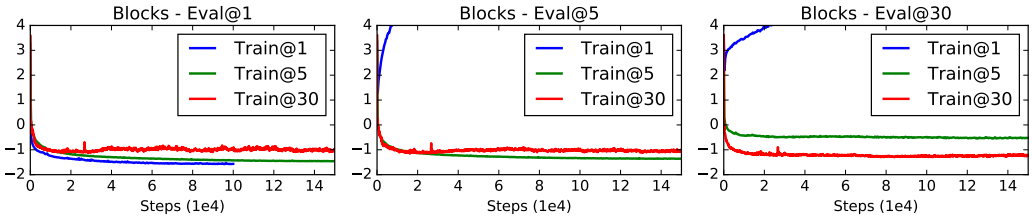

Figure 6: Each plot shows the validation performance as evaluated on predictions with different horizons. Within each plot we see the curves for models trained using different horizons. These curves show the importance of training for long horizons if we want to make long horizon predictions.

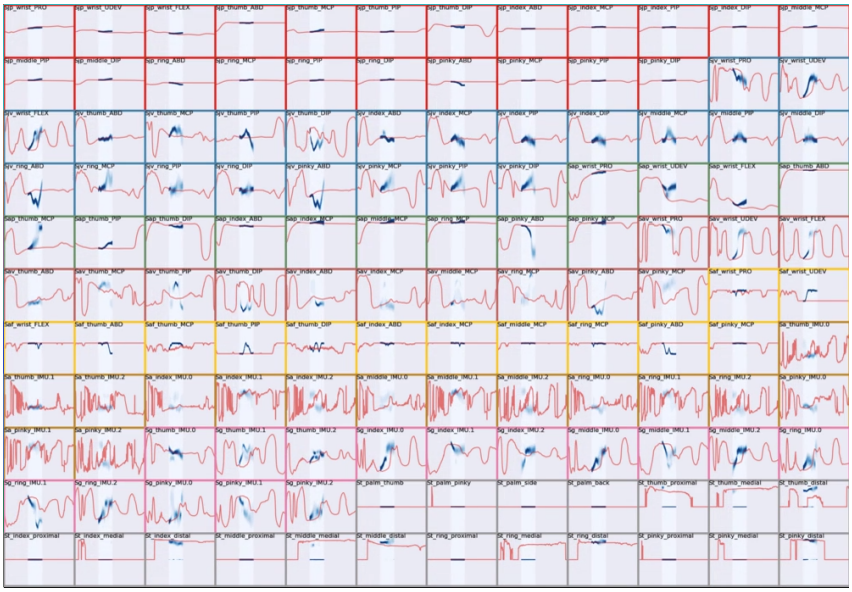

Figure 7: A visualization of the model planning to maximize predicted fingertip pressure.

A EXTRA EXPERIMENTS

A.1 USEFULNESS OF MULTI-STEP LIKELIHOODS

In this experiment we demonstrate the importance of training with multi-step likelihoods (rather than simply next step). We train models using 1, 5 and 30 step likelihoods during training, and compare each of their performance when making 1, 5 and 30 step predictions. The results of this experiment are shown in Figure 6 and clearly show that long horizon prediction during training is important for making accurate long horizon predictions at test time. Based on these results we use 30 step horizons for training throughout the rest of this section, since accurate predictions many steps into the future are important for our control experiments.

For the comparison in Figure 6 we did three separate parameter searches, one for each rollout length. We choose the model for the curves labelled `@n` by selecting the model with the best evaluation likelihood among models trained with that overshooting horizon, evaluated on rollouts of length n. The train `@n` eval `@m` curves show the model selected for good performance at n steps, evaluated for performance when predicting m steps.

Figures 7 and 8 show planned trajectories and model predictions when attempting to maximize fingertip pressure and to minimize predicted entropy, respectively.

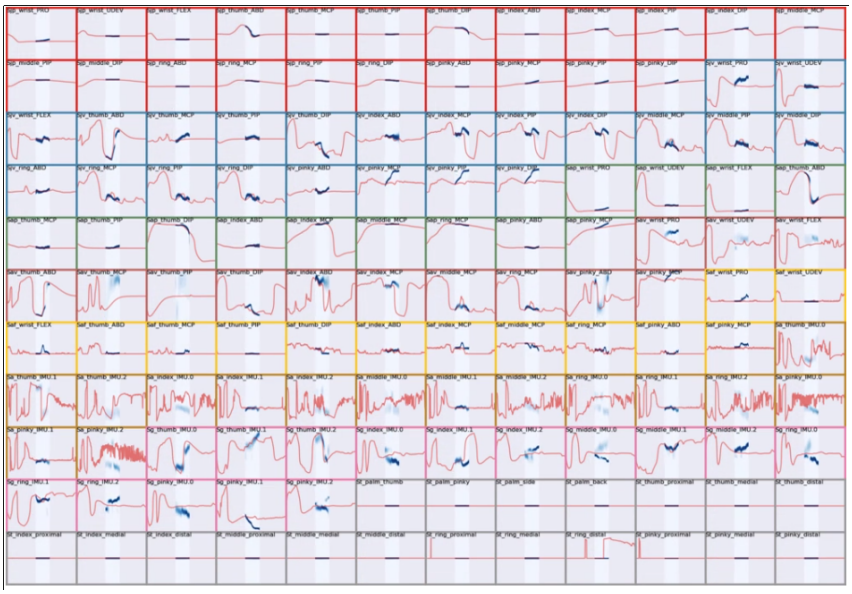

Figure 8: A visualization of the model planning to minimize predicted entropy.

B MPL HAND

The MPL hand is actuated by 13 motors each capable of exerting a bidirectional force on a single degree of freedom of the hand model. Each finger is actuated by a motor that applies torque to the MCP joint, and the MCP joint of each finger is coupled by a tendon to the PIP and DIP joints of the same finger, causing a single action to flex all joints of the finger together. Abduction of the main digits (ABD) is controlled by two motors attached to the outside of the index and pinky fingers, respectively. Unlike the main digits, the thumb is fully actuated, with separate motors driving each joint. The thumb has its own abduction joint, and somewhat strangely the thumb is composed of three jointed segments (unlike a human thumb which has only two). Each segment is separately controlled for a total of four actuators controlling the thumb. Finally the hand is attached to the world by fully actuated three three degree of freedom wrist joint, for a total of 13 actuators.

The hand model includes several sensors which we use as proprioceptive information. We observe the position and velocity of each joint in the model (three joints in the wrist and four in each finger except the middle which has no abduction joint, for a total of 22 joints), as well as the position, velocity and force of each of the 13 actuators. We also record from inertial measurement units (IMUs) located in the distal segment of each of the five fingers. Each IMU records three axis rotational and translational acceleration for a total of 30 acceleration measurements. Finally there are 19 pressure sensors placed throughout the inside of the hand that measure the magnitude of contact forces. Each finger including the thumb has three touch sensors, one on each segment (recall that the thumb has three segments in this model), and the palm of the hand has four different touch sensors that cover different regions. In total these sensors give a 132 dimensional proprioceptive state.

C HYPERPARAMETERS

All the hyperparameters reported in this section were found through random search, so the numbers are quite particular (e.g. 117 instead of a more natural 128), but the particularity should not be taken as a sign of delicacy.

control_embed_size	487
control_embed_hidden_size	146
control_embed_depth	1
sensor_embed_size	952
sensor_embed_hidden_size	N/A
sensor_embed_depth	0
encoder_hidden_size	496
decoder_hidden_size	496
likelihood_num_components	3
likelihood_mixture_hidden_size	815
likelihood_mixture_depth	2
mean_depth	2
mean_hidden_size	703
stddev_depth	0
stddev_hidden_size	N/A
num_sensors	132
learning_rate	0.000136554
num_overshoot_steps	30

Table 1: Hyperparameters for the experiments in Section 6.

control_embed_size	104
control_embed_hidden_size	60
control_embed_depth	2
sensor_embed_size	0
sensor_embed_hidden_size	N/A
sensor_embed_depth	51
encoder_hidden_size	117
decoder_hidden_size	117
likelihood_num_components	2
likelihood_mixture_hidden_size	N/A
likelihood_mixture_depth	0
mean_depth	0
mean_hidden_size	N/A
stddev_depth	1
stddev_hidden_size	76
num_sensors	53
learning_rate	8.82352e-05
num_overshoot_steps	10

Table 2: Hyperparameters for the experiments in Section 8.

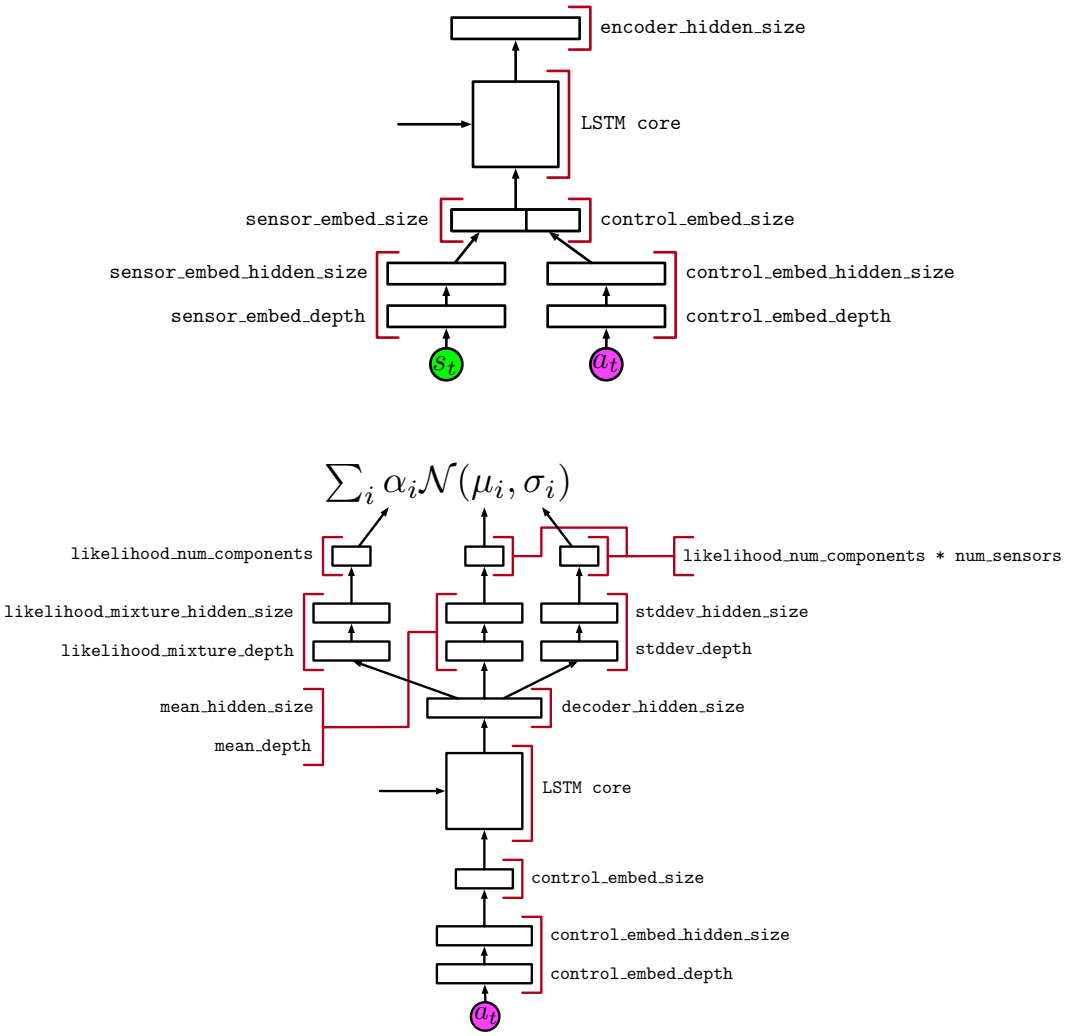

Figure 9: Detailed diagrams of the encoder and decoder modules, showing the structure of a single timestep of each model, along with labels that indicate different hyperparameters. A MLP sections of the models are parameterised with a depth and a hidden size, where a depth of d and a hidden size of k indicates d hidden layers of size k. We do not count the output layer or the input layer in the depth parameter (so a depth of 0 is a single linear transform followed by an activation function). The output layers of the MLP parts of the model are all indicated separately in the diagrams.