1 PART 1 : Introduction and Motivation

Using a model of the world to make decisions is the name of the game in model based reinforcement learning. It has found promising results in self-driving cars, control of robotics, operation results, games like chess, go etc.

1.1 Different approaches of models and learning models and trade offs:

- Dynamical Models : You derive the true differential equations of the world and use it to do control. This is the best thing one can do, if possible ofcourse.
- MLP, GNNs and locally linear learned approximations : If you have access to low level states, but can't figure out accurate dynamical equations then use these approaches. When state space is less than 8 dimensions, Gaussian Processes should be tried (PILCO).
- Observation space planning : Never do this!
- State Space Models : Infer latent states to do fast rollouts in imagination. it can be used in many ways like synthetic data generation.
- Recurrent value models (value equivalent models) : Learn representation only to predict the future bellman updates.

1.2 Some Questions, which the presentors will address in detail later, brought out good points about problems with MBRL :

- How does one decide the latent state dimensions.
- How does policy learning/planning suffer from model errors.
- For robotics, can a model learnt from a simulator(like mujoco) be for planning in real world? How robust are the policies learnt from such a model in the real world
- How to deal with inherent uncertainty of the world? Why to waste resources to predict things which are random over time.
- Single-step prediction models will result in compounding errors? How to deal with this?

1.3 Papers from this section and why would one want to read them(these are single line description, read the papers if one wants to explore that direction):

- [1] : Inferring unknown parameters of know dynamic equations or their approximations from observations. Like parameters of bicycle model of a vehicle
- [2] : Represent the relation between state variables as a graph and process them with GNNs.
- [3] : Idea of learning locally linear models.
- [4] [5] : Structured Latent state-transitions through object based representations.

1.4 Other takeaways from this section

- Data Augmentation has worked really well for model free RL, one might want to explore this for model learning as well.

2 PART 2 : Model-based Control

How can models be used in reinforcement learning [Figure 1]. The figure sums planning methods pretty well.

2.1 Standard terminology

Model : Any things that takes in the current environment state and can be used to predict the effects of action on the environment. Planning : A computation that uses the model to output an improved policy or sequence of actions [6].

Decision time planning : Using the model to make decisions on the go. Specialised methods like tree search [7] or ilqr are used for discrete and continuous actions spaces.
Background planning : learn reactive policies which are trained on data. Discrete and continuous actions kind of come under the same umbrella.

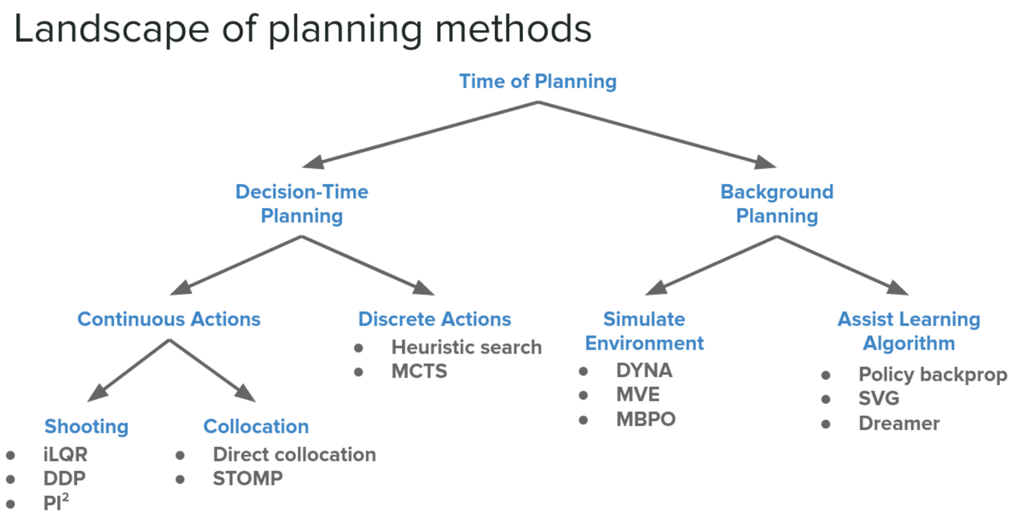

Figure 1: planning

It should be noted that these two methods can be mixed with each other to come up with a hybrid planning. For eg : you could learn robust reactive sub-policies for certain sub-tasks and make decision time plans over these sub-policies.

2.2 How are models used

- Simulating the environment : Mix trajectories from learnt model and real model to do model free reinforcement learning (Q-learning or Policy Gradients). eg DYNA-Q, MBPO(very reliable mbrl method), dreamer-v2.
- Assisting the learning algorithm : Use model to pass gradients and do end to end learning. eg dreamer-v1, value gradients etc.
- Strengthening the policy.

2.3 Questions from this section

- How to differentiate between the performance of representation learning vs model building and planning. When does planning help?
- Are there any theoretical guarantees to support mbrl methods?

2.4 Papers from this section and why would one want to read them

- [8] : Recent (2021) paper to understand the literature and where exactly planning seems to help.
- [9] : This paper has bounds on policy learning from data from a learnt model.

3 PART 3 : Model-based Control in the Loop

Can we learn model and improve policy iteratively?

3.1 Where should the data to learn the model come from?

Human demonstrations or sub-optimal policies or manually engineered policies. Model based offline RL has some recent work [10] [11] [12].

3.1.1 Train the data on its own model???

Data augmentation techniques, GANS and meta learning have been used for this recently.

Figure 2: planning

3.2 Can we learn from imperfect models?

Models wont be perfect cause training experience wont be diverse enough. Small errors might compound to become huge over longer time-scales and the planner might exploit these errors to provide values which are not really possible.

Some solutions:

- Do close loop control, continually re-plan. Expensive but there are workaround tricks in the paper [13] [14].
- If your algorithms estimate uncertainty of the model then plan conservatively. Stay close to certain trajectories.

3.3 Why do we need model uncertainty and how to estimate it?

Figure 2 shows two types of uncertainty in the world. Estimating uncertainty is an active area of DL research. Popular way is to train ensembles of small NNs independently to estimate uncertainty.

3.4 How to combine background and decision-time planning?

- Distillation : We store start states and successful decision planned trajectories (distillation dataset). Then learn a policy using behaviour cloning. [15] [16]
- Planning horizon is finite for trajectory optimisation (greedy behaviour). Add a final learnt value function to the cost of mpc : [17]. Even Muzero does this for discrete action space.
- Use planning as policy improvement : Recent paper by Jess Hamrick [8] gives a good overview.
- Implicit planning (planner inside policy) :Differentiable planning needs to be done : Differentiable MPC [18], Differentiable CEM [19], Control Oriented MBRL [20]. Value Prediction Networks, Value Iteration Networks etc.

3.5 Questions from this section

- Training for values wont capture true dynamics and would ofcourse be task specific AKA Problems with Value Equivalent Models. Outstanding answer by jessica : Yes, there is trade off for complicated problems which require sophisticated planning learning a value is a good idea. You could have a perfect model for the game go, but naive tree search wont get you anywhere.
- How to stay close to certain region? Gaussian process or NN ensembles capture uncertainty.
- Learning good latent representation through expert data : Model based offline learning is a new field to look at. [11] [12].
- Is there a reason why mpc is used so much in real world? Robustness to errors not compounding and MPC is not THAT costly.

4 PART 4 : Beyond Vanilla MBRL

What additional things can be done with a good model of the world:

4.1 Exploration

- Resettability : One can reset from any desirable state rather than just the start state [21].
- Intrinsic-reward based explorations : Create intrinsic rewards to completely explore the state-space [22]. Or even plan according to expected uncertainty [23](disagreement across transition functions - robust and thorough model of the world), [17] (disagreement accross value functions).

4.2 Representation Learning

Adding auxillary losses to create robust policies which are only used for learning desirable latent space.

- Using self supervised representation learning [24].
- Learning representations or abstractions which are easier to plan with. [25]. See Learning to drive using a model on rails [26].

4.3 Generalisation

I think this is the most crucial, how can MBRL methods help in adapting towards change in transition dynamics. Hamrick says that it is difficult(slow) to adapt a policy to change in dynamics or rewards.

- Adapting to change in rewards/goals. The model of the environment won't change if you have to perform a different task in the same environment. [27].
- Adapting to change in dynamics. If the real world is slightly different in certain cases (of course it will be) than your trained model. Meta-learning approach to adapt to changes at test time : [28].

5 Going forward in Model based RL

What qualities should a desirable model have? Which would lead to robust real world application.

- Faster Planning : Compositionality and Causality.
- High Tolerance to model error : Incompleteness and Adaptivity.
- Scalability to harder problems : Efficiency and Abstraction.

Hamrick explains all these aspects in a much better way than I can right now. Watch from 18:30 to 32:00.

Survey paper on model based RL : [29] has sections about safety, interpretability as well.

References

[1] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/d09bf41544a3365a46c9077ebb5e35c3-Paper.pdf`.

[2] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control, 2018.

[3] Michael C. Yip and David B. Camarillo. Model-less feedback control of continuum manipulators in constrained environments. *IEEE Transactions on Robotics*, 30(4):880–889, 2014. doi:10.1109/TRO.2014.2309194.

[4] Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P. Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration, 2019.

[5] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019.

[6] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

[7] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, and et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, Dec 2020. ISSN 1476-4687. doi:10.1038/s41586-020-03051-4. URL `http://dx.doi.org/10.1038/s41586-020-03051-4`.

[8] Jessica B. Hamrick, Abram L. Friesen, Feryal Behbahani, Arthur Guez, Fabio Viola, Sims Witherspoon, Thomas Anthony, Lars Buesing, Petar Veličković, and Théophane Weber. On the role of planning in model-based deep reinforcement learning, 2021.

[9] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization, 2019.

[10] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[11] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel : Model-based offline reinforcement learning, 2021.

[12] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization, 2020.

[13] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017. doi:10.1109/ICRA.2017.7989202.

[14] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012. doi:10.1109/IROS.2012.6386025.

[15] Igor Mordatch, Kendall Lowrey, Galen Andrew, Zoran Popovic, and Emanuel Todorov. Interactive control of diverse complex characters with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3132–3140, Cambridge, MA, USA, 2015. MIT Press.

[16] Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search, 2017.

[17] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control, 2019.

[18] Brandon Amos, Ivan Dario Jimenez Rodriguez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable mpc for end-to-end planning and control, 2019.

[19] Brandon Amos and Denis Yarats. The differentiable cross-entropy method, 2020.

[20] Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation, 2021.

[21] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems, 2021.

[22] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[23] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models, 2020.

[24] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks, 2016.

[25] Dane Corneil, Wulfram Gerstner, and Johanni Brea. Efficient model-based deep reinforcement learning with variational state tabulation, 2018.

[26] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails, 2021.

[27] Kevin Lu, Igor Mordatch, and Pieter Abbeel. Adaptive online planning for continual lifelong learning, 2020.

[28] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning, 2019.

[29] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. Model-based reinforcement learning: A survey, 2021.