A Unified View on Solving Objective Mismatch in Model-Based Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Model-based Reinforcement Learning (MBRL) aims to make agents more sample-efficient, adaptive, and explainable by learning an explicit model of the environment. While the capabilities of MBRL agents have significantly improved in recent years, how to best learn the model is still an unresolved question. The majority of MBRL algorithms aim at training the model to make accurate predictions about the environment and subsequently using the model to determine the most rewarding actions. However, recent research has shown that model predictive accuracy is often not correlated with action quality, tracing the root cause to the *objective mismatch* between accurate dynamics model learning and policy optimization of rewards. A number of interrelated solution categories to the objective mismatch problem have emerged as MBRL continues to mature as a research area. In this work, we provide an in-depth survey of these solution categories and propose a taxonomy to foster future research.

1 Introduction

Reinforcement learning (RL) has demonstrated itself as a promising tool for complex optimization landscapes and the creation of artificial agents by exceeding human performance in games (Mnih et al., 2015; Silver et al., 2016), discovering computer algorithms (Fawzi et al., 2022), managing power plants (Degrave et al., 2022), and numerous other tasks. The premise of RL is that complex agent behavior and decisions are driven by a desire to maximize cumulative rewards in dynamic environments (Silver et al., 2021). RL methods focus on learning a reward-optimal policy from sequences of state-action-reward tuples. These methods can be broadly classified as model-free RL (MFRL) and model-based RL (MBRL). MFRL methods directly learn the policy from the environment samples, whereas MBRL approaches learn an explicit model of the environment and use the model in the policy-learning process. MBRL methods are advantageous because they can make deep RL agents more sample-efficient, adaptive, and explainable. Prior work has shown that MBRL methods allow agents to plan with respect to variable goals or diverse environments (Zhang et al., 2018; Hafner et al., 2023) and that designers can introspect an agent's decisions (van der Waa et al., 2018), which can help in identifying potential causes for failures (Räuker et al., 2023).

Despite the benefits of MBRL, there is considerable divergence in existing algorithms and no consensus on the aspects of the environment to model and how the model should be learned (e.g., model architecture and data arrangement). For example, Dyna-style MBRL algorithms are trained to make accurate predictions about the environment, then find the optimal actions or policy with respect to the trained model (Sutton & Barto, 2018). The intuition behind these approaches is that improving the model's accuracy in predicting environment dynamics should facilitate better action selection and improved algorithm performance. However, recent research found that improved model accuracy often does not correlate with higher achieved returns (Lambert et al., 2020). While the underperformance of policies trained on the learned models is often due to the models' inability to sufficiently capture environment dynamics and the policy exploiting errors in the model (Jafferjee et al., 2020), Lambert et al. (2020) attributed the root cause of this phenomenon to the *objective mismatch* between model learning and policy optimization: while the policy is trained to maximize return, the model is trained for a different objective and not aware of its role in the policy decision-making

Algorithm 1 Basic algorithm of Model-based reinforcement learning

Require: Environment, model $\hat{P}$, policy π, data buffer $\mathcal{D}$, time budget T
  while $t \leq T$ **do**
    Interact with the environment and collect data $\mathcal{D} \leftarrow \mathcal{D} \cup (s, a, r, s')$
    Update model $\hat{P}$
    Update policy π
  end while

process. This objective mismatch problem represents a substantial and fundamental limitation of MBRL algorithms, and resolving it will likely lead to enhanced agent capabilities.

In this review, we study existing literature and provide a unifying view of different solutions to the objective mismatch problem. Our main contribution is a taxonomy of four categories of decision-aware MBRL approaches: *Distribution Correction, Control-As-Inference, Value-Equivalence, and Differentiable Planning*, which derive modifications to the model learning, policy optimization, or both processes for the purpose of aligning model and policy objectives and gaining better performance (e.g., achieving higher returns). For each approach, we discuss its intuition, implementation, and evaluations, as well as implications for agent behavior and applications. This review is complementary to prior broader introductions of MBRL (e.g., see Moerland et al. (2023); Luo et al. (2022)) in its in-depth analysis of solutions to the objective mismatch problem and illustrations of implications for MBRL approaches.

2 Background

To facilitate comparisons between the reviewed approaches, we adopt the common notation and premise for MBRL based on (Sutton & Barto, 2018). In the subsequent sections, we introduce Markov Decision Processes, reinforcement learning, and the objective mismatch problem.

2.1 Markov Decision Process

We consider reinforcement learning in Markov Decision Processes (MDP) defined by tuple $(\mathcal{S}, \mathcal{A}, P, R, \mu, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the environment transition probability function (also known as the dynamics model), $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mu : \mathcal{S} \to \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. The RL agent interacts with the environment using a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ and generates trajectories $\tau = (s_{0:T}, a_{0:T})$ distributed according to $P(\tau)$ and evaluated by discounted cumulative rewards (also known as return) $R(\tau)$. $P(\tau)$ and $R(\tau)$ are defined respectively as:

$$P(\tau) = \mu(s_0) \prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\pi(a_t|s_t), \quad R(\tau) = \sum_{t=0}^{T} \gamma^t R(s_t, a_t)\,. \tag{1}$$

Abusing notation, we also use $P(\tau)$ and $R(\tau)$ to refer respectively to the probability measure and discounted reward for infinite horizon sequences $\tau = (s_{0:\infty}, a_{0:\infty})$. We further denote the marginal state-action density of policy π in the environment P (also known as the normalized occupancy measure) as $d_P^\pi(s, a) = (1 - \gamma)\mathbb{E}_{P(\tau)}[\sum_{t=0}^{\infty} \gamma^t P(s_t = s, a_t = a)]$.

2.2 Reinforcement Learning

The goal of the RL agent is to find a policy that maximizes the expected return $J_P(\pi)$ in the environment with dynamics P:

$$\max_\pi J_P(\pi) = \mathbb{E}_{P(\tau)}[R(\tau)]\,. \tag{2}$$

Importantly, the agent does not know the true environment dynamics and has to solve (2) without this knowledge. We assume the reward function is known.

(2) is often solved by estimating the state value function $V(s)$ and state-action value function $Q(s,a)$ of the optimal policy using the Bellman equation:

$$\begin{aligned} Q(s,a) &= R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V(s')]\,, \\ V(s) &= \max_{a \in \mathcal{A}} Q(s,a)\,. \end{aligned} \tag{3}$$

Then the optimal policy can be constructed by taking actions according to:

$$\pi(a|s) \in \arg\max_{a \in \mathcal{A}} Q(s,a)\,. \tag{4}$$

Variations to (3) and (4) exist depending on the problem at hand. For example, in continuous action space, the maximum over action is typically found approximately using gradient descent (Lillicrap et al., 2015; Schulman et al., 2017a). Policy iteration algorithms estimate the value of the current policy by defining $V(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q(s,a)]$ and improve upon the current policy. We refer to the process of finding the optimal policy for an MDP, including these variations, as *policy optimization*.

In model-free RL, the expectation in (3) is estimated using samples from the environment. Model-based RL instead learns a model $\hat{P}$ and estimates the value function using samples from the model (often combined with environment samples). These algorithms alternate between data collection in the environment, updating the model, and improving the policy (see Algorithm 1). We will be mostly concerned with forward dynamics model of the form $\hat{P}(s'|s,a)$, although other types of dynamics models such as inverse dynamics models, can also be used (Chelu et al., 2020).

2.3 Model Learning and Objective Mismatch

Many MBRL algorithms train the model $\hat{P}$ to make accurate predictions of environment transitions. This is usually done via maximum likelihood estimation (MLE):

$$\max_{\hat{P}} L_{\text{MLE}}(\hat{P}) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}} \left[\log \hat{P}(s'|s,a) \right]\,. \tag{5}$$

This choice is justified by the well-known simulation lemma (Kearns & Singh, 2002), which was further refined in recent state-of-the-art MBRL algorithms to account for off-policy learning:

Theorem 2.1. *(Lemma 3 in (Xu et al., 2020)) Given an MDP with bounded reward:* $\max_{s,a} |R(s,a)| = R_{max}$ *and dynamics model* P*, a data-collecting behavior policy* π_b*, and a learned model* $\hat{P}$ *with* $\mathbb{E}_{(s,a)\sim d_P^{\pi_b}} D_{KL}[P(\cdot|s,a)||\hat{P}(\cdot|s,a)] \leq \epsilon_P$*, for arbitrary policy* π *with bounded divergence* $\epsilon_\pi \geq \max_s D_{KL}[\pi(\cdot|s)||\pi_b(\cdot|s)]$*, the policy evaluation error is bounded by:*

$$|J_{\hat{P}}(\pi) - J_P(\pi)| \leq \frac{\sqrt{2}\gamma R_{\max}}{(1-\gamma)^2}\sqrt{\epsilon_P} + \frac{2\sqrt{2}\gamma R_{\max}}{(1-\gamma)^2}\sqrt{\epsilon_\pi}\,. \tag{6}$$

Thus, one way to reduce the policy evaluation error of the optimizing policy π is to make the model as accurate as possible in state-action space visited by the behavior policy while maintaining small policy divergence. There are two issues with this approach. First, unlike the tabular setting, the dynamics error might not reduce to zero even with infinite data in complex, high dimensional environments due to limited model capacity or model misspecification. Second, maintaining small policy divergence in the second term requires knowledge of the behavior policy in all states and additional techniques for constrained policy optimization. The former requirement can be demanding with an evolving behavior policy as in most online RL settings, or with an unknown behavior policy as in most offline RL settings.

Towards better understanding of model learning and policy performance, Lambert et al. (2020) found that model predictive accuracy is often not correlated with the achieved returns. One of the main reasons for this finding is that the dynamics model learned by one-step prediction can be inaccurate for long-horizon rollouts due to compounding error (Lambert et al., 2022). These inaccuracies can then be exploited during policy optimization (Jafferjee et al., 2020). Various methods have been proposed to improve the dynamics model's

long-horizon prediction performance or avoid exploitation by the policy, for example by using multistep objectives (Luo et al., 2018), training the model to self-correct (Talvitie, 2017), directly predicting the marginal density (Janner et al., 2020), training a model with different temporal abstractions (Lambert et al., 2021), or quantifying epistemic uncertainty in the model (Chua et al., 2018). Nevertheless, a fundamental issue remains that the model is trained with an objective misaligned with the final return of the policy.

2.4 Towards Decision-Aware MBRL

Overcoming the objective mismatch problem has important implications for safe and data-efficient RL. In domains where environment interaction is expensive or unsafe, off-policy or offline RL are used to extract optimal policies from a limited dataset (Levine et al., 2020). In offline MBRL, the dynamics model is typically fixed after an initial pretraining stage and other methods are required to prevent model-exploitation from the policy, such as by designing pessimistic penalties (Yu et al., 2020b; Kidambi et al., 2020). Decision-aware MBRL has the potential to simplify or remove the design of these post hoc methods.

Beyond data-efficiency and safety, decision-aware MBRL can potentially address the gaps in current automated decision-making software systems in various domains, such as transportation and health care (McAllister et al., 2022; Wang et al., 2021). These systems are traditionally developed and tested in a modular fashion. For example, in automated vehicles, the trajectory forecaster is typically developed independently of the vehicle controller. As a result, modules which pass the unit test may still fail the integration test.

Thus, the core motivation for solving the objective mismatch problem is to improve MBRL agent capabilities and downstream performance by designing model learning objectives that are aware of its role in or directly contribute to policy optimization, policy objectives that are aware of model deficiencies, or unified objectives that contribute to both (Lambert et al., 2020). It is therefore desirable if guiding principles can be identified to facilitate the design of future objectives. The goal of this survey is to identify these principles by synthesizing existing literature.

3 Survey Scope and Related Work

The focus of this survey is on existing approaches that address the objective mismatch problem. We identified these approaches by conducting a literature search of the terms "objective mismatch," "model-based," and "reinforcement learning" and their permutations on Google Scholar and Web of Science. We compounded these searches by examining articles citing (Lambert et al., 2020) and their reference lists. We focused on (Lambert et al., 2020) because it was the first work to identify and name the objective mismatch problem. The initial search yielded 82 results which were screened for relevance. To be included in the survey, an article had to specifically address a solution to the objective mismatch problem or propose decision-aware objectives for model learning and policy optimization. Articles that simply discussed the objective mismatch problem without providing a solution were excluded. After abstract and full text screening, we retained a total of 43 papers.

Before presenting the included papers, we briefly cover related research areas that are deemed out-of-scope by our screening criteria in a non-exhaustive manner; these include state abstraction, representation learning, control-aware dynamics learning, and decision-focused learning. MDP state abstraction focuses on aggregating raw state features while preserving relevant properties of the dynamics model, policy, or value function (Li et al., 2006; Abel et al., 2020). For example, bisimulation-based aggregation methods aim to discover models with equivalent transitions and have been applied in the deep RL setting (Zhang et al., 2020). Recent works on representation learning for control aim to factorize controllable representations from uncontrollable latent variables using, for example, inverse dynamics modeling (Lamb et al., 2022). These research directions are related to the objective mismatch problem in that they also aim to overcome the difficulty of modeling complex and irrelevant observations for control, however, they are out-of-scope for the current survey because they do not consider the mutual adaptation of model and policy while learning. Control-aware dynamics learning focus on dynamics model regularization based on smoothness or stabilizability principles in order to address compounding error in synthesizing classic (e.g., LQR) or learned controllers (Levine et al., 2019; Singh et al., 2021; Richards et al., 2023). However, they do not directly focus on the source of the problem,

i.e., misaligned model and policy objectives, and thus are out-of-scope for the current survey. Objective mismatch is also related to the broad topic on decision-focused learning, where a model is trained to predict the parameters of an optimization problem, which is subsequently solved (Wilder et al., 2019; Elmachtoub & Grigas, 2022). Due to the focus on sequential decision-making problems (i.e., RL), more general work on decision-focused learning is out-of-scope.

4 Taxonomy

Our synthesis of the 43 papers identified 4 broad categories of approaches to solving the objective mismatch problem. We consolidated these into the following taxonomy of decision-aware MBRL:

- **Distribution Correction** adjusts for the mismatched training data in both model learning and policy optimization.
- **Control-As-Inference** provides guidance for the design of model learning and policy optimization objectives by formulating both under a single probabilistic inference problem.
- **Value-Equivalence** searches for models that are homomorphic to the true environment and value function.
- **Differentiable Planning** embeds the model-based policy optimization process in a differentiable program such that both the policy and the model can be optimized towards the same objective.

A schematic of the relationships between these approaches is shown in Figure 1. The figure illustrates that Value-Equivalence approaches are the most prevalent in the literature, however, more recent approaches have concentrated on distribution correction and control-as-inference. The remaining sections discuss these approaches with a focus on their model learning and policy optimization objectives. Comparisons of the design and evaluations of the core reviewed algorithms are provided in Table 1 and Table 2, respectively.

4.1 Distribution Correction

Distribution correction aims to correct for training errors attributable to policy optimization on samples not from the true environment (model-shift) or to samples from an outdated or different policy (policy-shift). Our literature search identified one approach focused on addressing model-shift and four approaches focused on policy-shift.

4.1.1 Addressing Model-Shift

Haghgoo et al. (2021) proposed to address model-shift by using the learned model as the proposal distribution for an importance sampling estimator of the expected return in the true environment as defined in (7):

$$\mathbb{E}_{P(\tau)}[R(\tau)] = \mathbb{E}_{Q(\tau)}\left[\frac{P(\tau)}{Q(\tau)}R(\tau)\right] = \mathbb{E}_{Q(\tau)}[w(\tau)R(\tau)]\,. \tag{7}$$

where $P(\tau)$ is the trajectory distribution in the true environment, $Q(\tau) = \mu(s_0)\prod_{t=0}^{\infty}\hat{P}(s_{t+1}|s_t,a_t)\pi(a_t|s_t)$ is the trajectory distribution induced by the learned model $\hat{P}$ (known as the proposal distribution in importance sampling), and $w(\tau)$ is the importance weight. Since importance sampling is agnostic to the proposal distribution (i.e., $Q(\tau)$), this method can be applied to models trained using any objectives.

Given that both $P(\tau)$ and $Q(\tau)$ correspond to the MDP structure, the importance weight can be decomposed over time steps as:

$$w(s_{0:t}, a_{0:t}) = \prod_{m=0}^{t-1}\frac{P(s_{m+1}|s_m,a_m)}{\hat{P}(s_{m+1}|s_m,a_m)} = \prod_{m=0}^{t-1} w(s_m, a_m, s_{m+1})\,. \tag{8}$$

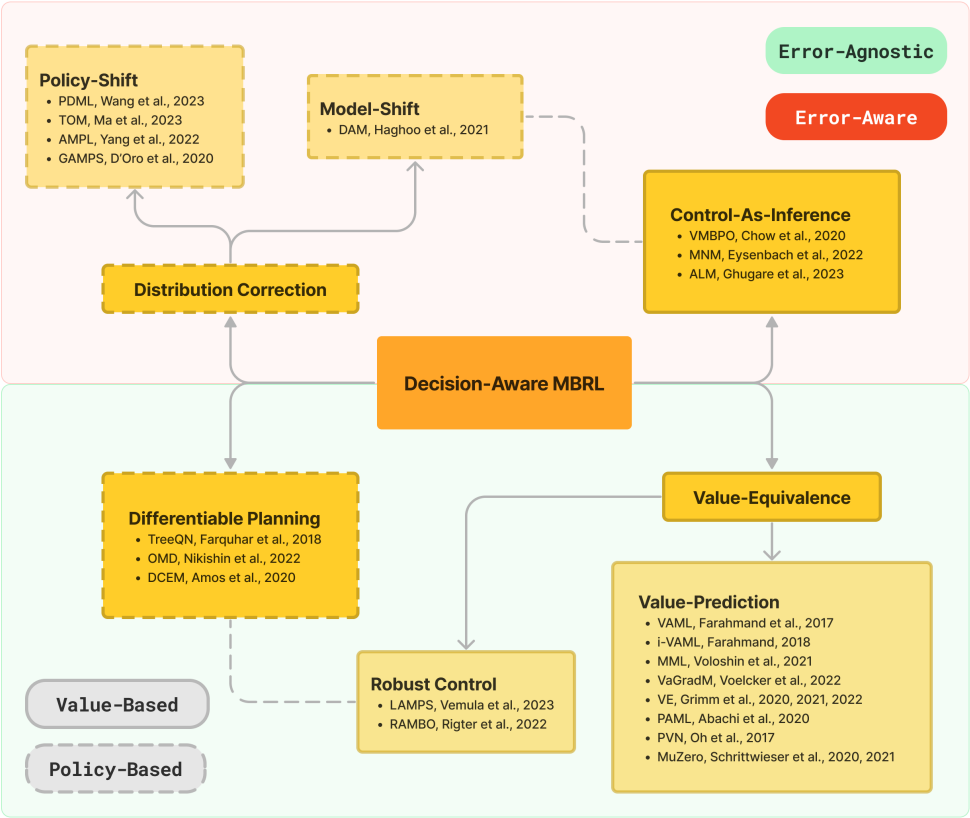

Figure 1: Schematic of the relationships between the core surveyed decision-aware MBRL approaches. Direct relationships are shown in solid arrows. Indirect relationships are shown in dashed connections. The algorithms in each category are sorted by the order in which they are presented in the paper.

Although not directly available, the per-time step importance weight $w(s_m, a_m, s_{m+1})$ can be obtained using density ratio estimation through binary classification, similar to Generative Adversarial Networks (Sugiyama et al., 2012; Goodfellow et al., 2020). The policy is then optimized with respect to the importance-weighted reward. Thus, the authors refer to this method as Discriminator Augmented MBRL (DAM).

To address the possibility that the estimator can have high variance for long trajectories due to multiplying the importance weights, the authors proposed to optimize towards the following model to achieve the lowest variance for the estimator, which is shown in (Goodfellow et al., 2016) to have the following form:

$$Q^*(\tau) \propto P(\tau)R(\tau)\,. \tag{9}$$

Such a model can be found by minimizing $D_{KL}[Q^*(\tau)||Q(\tau)]$, resulting in a return-weighted model training objective. DAM's model learning and policy optimization objectives are defined as follows:

$$\max_{\hat{P}} L_{\text{DAM}}(\hat{P}) = \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty} R(\tau)\log \hat{P}(s_{t+1}|s_t, a_t)\right]\,, \tag{10}$$

$$\max_{\pi} L_{\text{DAM}}(\pi) = \mathbb{E}_{Q(\tau)}\left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{m=0}^{t} w(s_m, a_m, s_{m+1})\right) R(s_t, a_t)\right]\,. \tag{11}$$

The authors showed that DAM outperforms the standard Dyna MBRL algorithm with a MLE model objective in custom driving environments with multi-modal environment dynamics. They further found that including the importance weights in the policy objective is crucial for its superior performance.

4.1.2 Addressing Policy-Shift

Rather than focusing on model-shift, the policy-shift approaches address objective mismatch by re-weighting the model training data such that samples less relevant or collected far away from the current policy's marginal state-action distribution are down-weighted.

Model training data in MBRL is typically collected by policies different from the current policy due to continuous policy updating. Wang et al. (2022) found that when training the model to make accurate predictions on all data, the model produces more error on the most recent data compared to its overall performance. To this end, they propose an algorithm called Policy-adapted Dynamics Model Learning (PDML) which stores all historical policies and uses an estimate of the divergence between the data-collecting policy and the current policy to weight data samples for MLE model training.

In line with Wang et al. (2022) to address inferior model predictions on updated policies, Ma et al. (2023) proposed a weighted model training scheme motivated by the following lower bound of the log-transformed expected return:

$$\begin{aligned}
\log J_P(\pi) &= \log \mathbb{E}_{d^\pi_{\hat{P}}(s,a,s')}[R(s,a)] \\
&= \log \mathbb{E}_{d^\pi_{\hat{P}}(s,a,s')}\left[\frac{d^\pi_P(s,a,s')}{d^\pi_{\hat{P}}(s,a,s')}R(s,a)\right] \\
&\geq \mathbb{E}_{d^\pi_{\hat{P}}(s,a,s')}\left[\log \frac{d^\pi_P(s,a,s')}{d^\pi_{\hat{P}}(s,a,s')} + \log R(s,a)\right] \\
&\geq -D_f[d^\pi_{\hat{P}}(s,a,s')||d^\pi_P(s,a,s')] + \mathbb{E}_{d^\pi_{\hat{P}}(s,a,s')}[\log R(s,a)]
\end{aligned} \tag{12}$$

where $d^\pi_P(s,a,s') = (1-\gamma)\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s, a_t = a, s_{t+1} = s')]$ denotes the normalized occupancy measure for transitions (s,a,s'), and D_f denotes a chosen f-divergence measure. The first term in the lower bound suggests that the learned dynamics model should induce similar transition occupancy to the true environment under the current policy. To this end, the authors proposed the Transition Occupancy Matching (TOM) algorithm to optimize the first term while alternating with optimizing the second term using model-based policy optimization (MBPO) (Janner et al., 2019).

While one could optimize the first term by simulating π in $\hat{P}$, this approach requires repeated simulation for each dynamics model update and fails to leverage data collected by the behavior policy. Instead, the authors proposed density ratio-corrected MLE model training and estimate the importance weight using dual-RL techniques (Nachum & Dai, 2020) as follows:

$$w_{\text{TOM}}(s,a,s') = \frac{d^\pi_{\hat{P}}(s,a,s')}{d^{\pi_b}_P(s,a,s')} = f'_\star\left(\log \frac{d^\pi_P(s,a,s')}{d^{\pi_b}_P(s,a,s')} + \gamma\mathbb{E}_{\pi(a'|s')}[\tilde{Q}(s',a')] - \tilde{Q}(s,a)\right). \tag{13}$$

where the log density ratio on the right hand side is estimated using a discriminator, $\tilde{Q}(s,a)$ is a dual value function capturing how easy it is to distinguish transitions generated by π and π_b in P, and $f'_\star$ is the derivative of the Fenchel conjugate of the chosen f-divergence function.

This formulation leads to the following model and policy objectives:

$$\begin{aligned}
\max_{\hat{P}} L_{\text{TOM}}(\hat{P}) &= \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[w_{\text{TOM}}(s,a,s')\log \hat{P}(s'|s,a)\right], \\
\max_{\pi} L_{\text{TOM}}(\pi) &= \mathbb{E}_{d^\pi_{\hat{P}}(s,a,s')}[\log R(s,a)].
\end{aligned} \tag{14}$$

Empirically, the authors showed that TOM correctly assigns higher importance weights w to transition samples more similar to the current policy and that TOM is more sample efficient and achieves higher asymptotic performance in MuJoCo environments (Todorov et al., 2012) than MBPO and PDML which uses heuristic policy divergence based weighting. The authors also compared TOM with VaGradM (a different class of decision-aware MBRL algorithm reviewed in section 4.3.1) and found TOM performs significantly better except for one environment.

In contrast to the above online RL approaches, Yang et al. (2022) addressed the offline RL setting and proposed the Alternating Model and Policy Learning (AMPL) algorithm to optimize the following lower bound of the expected return:

$$\begin{aligned} J_P^\pi &\geq J_{\hat{P}}^\pi - |J_P^\pi - J_{\hat{P}}^\pi| \\ &\geq J_{\hat{P}}^\pi - \frac{\gamma R_{max}}{\sqrt{2}(1-\gamma)} \sqrt{D_\pi(P, \hat{P})} \end{aligned} \tag{15}$$

where the divergence in the second term which characterizes policy-shift is defined as:

$$D_\pi(P, \hat{P}) = \mathbb{E}_{(s,a)\sim d_P^{\pi_b}} \left[w_{\text{AMPL}}(s,a) D_{KL}[P(s'|s,a)\pi_b(a'|s') || \hat{P}(s'|s,a)\pi(a'|s')] \right] \tag{16}$$

and $w_{\text{AMPL}}(s,a) = \frac{d_P^\pi(s,a)}{d_P^{\pi_b}(s,a)}$ is the marginal state-action density ratio between current policy and the behavior policy.

Similar to TOM, the bound in (15) suggests MLE dynamics model learning weighted by $w_{\text{AMPL}}(s,a)$. However, for policy training, (15) suggests not only maximizing reward but also minimizing the divergence so that the evaluation error in the second term is controlled for. Instead of using dual-RL techniques to estimate w as in TOM, the authors proposed a novel fixed point estimator. They further approximated $D_\pi(P, \hat{P})$ for policy optimization using the output of a discriminator $\log(1 - C(s,a))$. The APML model and policy objectives are thus defined as follows:

$$\begin{aligned} \max_{\hat{P}} L_{\text{APML}}(\hat{P}) &= \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[w_{\text{AMPL}}(s,a) \log \hat{P}(s'|s,a)] \,, \\ \max_{\pi} L_{\text{APML}}(\pi) &= \mathbb{E}_{d_{\hat{P}}^\pi}[R(s,a) - \log(1 - C(s,a))] \,. \end{aligned} \tag{17}$$

A similar method was proposed in (Hishinuma & Senda, 2021) which instead estimates the importance weight using full-trajectory model rollouts.

Empirically, the authors showed that APML outperforms a number of SOTA model-based and model-free offline RL approaches (e.g., MOPO; (Yu et al., 2020b), COMBO (Yu et al., 2021), CQL (Kumar et al., 2020)) in most D4RL environments (Fu et al., 2020) and that including the importance weight w in the model objective is crucial for its performance.

Also in the offline RL setting, D'Oro et al. (2020) focused specifically on policy-gradient estimation using samples from the offline dataset, but the value function is estimated using the learned model (model-value gradient; MVG). This setup can be advantageous since the model bias is limited to the value estimate but not the sample-based evaluation of expectation. They proposed the Gradient-Aware Model-based Policy Search (GAMPS) algorithm which trains the model to minimize the following bound on the difference from the true policy-gradient:

$$\|\nabla_\pi J_P(\pi) - \nabla_\pi J^{\text{MVG}}(\pi)\| \leq \frac{\gamma\sqrt{2} Z R_{max}}{(1-\gamma)^2} \sqrt{\mathbb{E}_{(s,a)\sim\eta_P^\pi} D_{KL}[P(\cdot|s,a) || \hat{P}(\cdot|s,a)]} \tag{18}$$

where Z is a constant and η_P^π is a state-action density weighted by the policy-gradient norm, which suggests a model training objective where samples with higher policy-gradient norms are up-weighted.

The GAMPS model and policy objectives are defined as follows:

$$\begin{aligned} \max_{\hat{P}} L_{\text{GAMPS}}(\hat{P}) &= \mathbb{E}_{\tau\sim\mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{m=0}^{t} \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)} \sum_{l=0}^{t} \|\nabla \log \pi(a_l|s_l)\| \right) \log \hat{P}(s_{t+1}|s_t, a_t) \right] \,, \\ \max_{\pi} L_{\text{GAMPS}}(\pi) &= \mathbb{E}_{\tau\sim\mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{m=0}^{t} \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)} \right) Q(s_t, a_t) \log \pi(a_t|s_t) \right] \,. \end{aligned} \tag{19}$$

Empirically, the authors first used a diagnostic gridworld environment to show that GAMPS can accurately estimate policy gradients with a constrained dynamics model class which conditions only on actions but not

on states. On two MuJoCo tasks, they showed that GAMPS achieved higher asymptotic performance than MLE model learning and two classical model-free RL algorithms adapted to the offline setting.

Distribution Correction Summary: All approaches in this category focused on return estimation under mismatched distributions. The single approach addressing *model-shift* adopted a straightforward importance sampling estimator, while the rest of the approaches have focused on bounding policy or policy-gradient evaluation error with respect to *policy-shift* as a result of deviating from the behavior policy. All approaches also adopted a weighted maximum likelihood model learning objective where the weights represent relevance to policy optimization, such as higher return, closeness to the current policy, or potential for policy improvement. However, by changing weights over the course of training, these models are able to adapt to changes in the current policy as opposed to being policy-agnostic as in standard MLE model learning.

4.2 Control-As-Inference

Besides the distribution correction approaches, other methods have attempted to leverage existing validated approaches to solve the objective mismatch problem. One such principled approach is to leverage the control-as-inference framework (Levine, 2018) and formulate both model learning and policy optimization as a single probabilistic inference problem.

The core concept of control-as-inference is that optimal control can be formulated as a probabilistic inference problem if we assume optimal behavior were to be observed in the future and compute a posterior distribution over the unknown actions that led to such behavior. Most control-as-inference methods define optimality using a binary variable $\mathcal{O}$ with the following distribution:

$$P(\mathcal{O}_t = 1|s_t, a_t) = \exp(R(s_t, a_t))\,. \tag{20}$$

The posterior, represented as a variational distribution $Q(\tau)$, is found by maximizing the lower bound of the marginal likelihood of optimal trajectories as follows:

$$\begin{aligned}
\log P(\mathcal{O}_{0:\infty} = 1) &= \log \int_\tau P(\mathcal{O}_{0:\infty}, \tau) \\
&= \log \mathbb{E}_{P(\tau)}[P(\mathcal{O}_{0:\infty} = 1|\tau)] \\
&= \log \mathbb{E}_{P(\tau)}\left[\exp\left(\sum_{t=0}^{\infty} R(s_t, a_t)\right)\right] \\
&\geq \mathbb{E}_{Q(\tau)}\left[\sum_{t=0}^{\infty} R(s_t, a_t)\right] - D_{KL}[Q(\tau)||P(\tau)]
\end{aligned} \tag{21}$$

The prior $P(\tau)$ is usually defined using the structure of the MDP:

$$P(\tau) = \mu(s_0) \prod_{t=0}^{\infty} P(s_{t+1}|s_t, a_t) P(a_t|s_t)\,. \tag{22}$$

where $P(a|s)$ is a prior or default policy usually set to a uniform distribution.

Most of the design decisions in control-as-inference are incorporated in defining the variational distribution $Q(\tau)$. Prior control-as-inference approaches, such as the ones reviewed in (Levine, 2018), define $Q(\tau)$ using the environment distribution and do not incorporate a learned model in the formulation. Thus, the majority of those algorithms are model-free.

In contrast to those approaches, Chow et al. (2020) proposed a joint model learning and policy optimization algorithm under the control-as-inference framework called Variational Model-Based Policy Optimization

(VMBPO) by incorporating the learned dynamics model in the variational distribution defined as follow:

$$Q(\tau) = \mu(s_0) \prod_{t=0}^{\infty} \hat{P}(s_{t+1}|s_t, a_t)\pi(a_t|s_t)\,. \tag{23}$$

Although $Q(\tau)$ has the same structure as the proposal distribution in the importance sampling-based algorithm DAM (see (7)), the model learning and policy optimization objectives of VMBPO are derived from the marginal likelihood lower bound $L(\hat{P}, \pi)$ defined as:

$$L_{\text{VMBPO}}(\hat{P}, \pi) = \mathbb{E}_{Q(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t) + \log \frac{P(a_t|s_t)}{\pi(a_t|s_t)} + \log \frac{P(s_{t+1}|s_t, a_t)}{\hat{P}(s_{t+1}|s_t, a_t)} \right) \right]\,. \tag{24}$$

Defining a Bellman-like recursive equation based on (24):

$$Q(s, a) = R(s_t, a_t) + \log \frac{P(a|s)}{\pi(a|s)} + \mathbb{E}_{\hat{P}(s'|s,a)} \left[V(s') + \log \frac{P(s'|s, a)}{\hat{P}(s'|s, a)} \right]\,. \tag{25}$$

The authors showed that the optimal variational dynamics and policy have the following form:

$$\begin{aligned} \hat{P}(s'|s, a) &\propto \exp(V(s') + \log P(s'|s, a))\,, \\ \pi(a|s) &\propto \exp(Q(s, a) + \log P(a|s))\,. \end{aligned} \tag{26}$$

Similar to DAM, the authors used a discriminator to estimate the dynamics density ratio.

Interestingly, the optimal dynamics are encouraged to be not only similar to the true dynamics via the log likelihood term, but also have a higher tendency to sample high value states. Such an optimistic dynamics model is similar to that of DAM, albeit for a different reason that the dynamics is conditioned on the optimality variable while performing variational inference. Also similar to DAM's importance weight-augmented reward, the policy optimizes not only reward but also the negative log density ratio between the learned and the ground truth dynamics. This encourages the policy to visit states where the learned dynamics is accurate (lower cross entropy with the ground truth dynamics) and also states where the learned dynamics is uncertain (higher entropy).

Empirically, the authors showed that VMBPO outperforms SOTA model-based (e.g., MBPO) and model-free baselines (e.g., SAC (Haarnoja et al., 2018)) in six MuJoCo environments for a range of policy learning rate settings. However, they did not evaluate specific agent properties discussed in the previous paragraph.

While Chow et al. (2020) was the first to propose a unified decision-aware objective for model learning and policy optimization, Eysenbach et al. (2022) pointed out that the objective (24) is an upper-bound on the RL objective, which can be decomposed as the expected return and its variance, and thus undesirable. They instead proposed an algorithm called Mismatched-No-More (MNM) with an alternative optimality variable:

$$P(\mathcal{O} = 1|\tau) = R(\tau) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\,. \tag{27}$$

Using this definition of the optimality variable, they derived a modified log marginal likelihood lower bound:

$$\begin{aligned} \log P(\mathcal{O} = 1) &= \log \int_{\tau} P(\mathcal{O} = 1, \tau) \\ &= \log \mathbb{E}_{P(\tau)}[P(\mathcal{O} = 1|\tau)] \\ &\geq \mathbb{E}_{Q(\tau)}[\log R(\tau) + \log P(\tau) - \log Q(\tau)] \\ &= \mathbb{E}_{Q(\tau)} \left[\log \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) + \log P(\tau) - \log Q(\tau) \right] \\ &\geq \mathbb{E}_{Q(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t \log R(s_t, a_t) + \log P(\tau) - \log Q(\tau) \right] \end{aligned} \tag{28}$$

The benefit of this modification is that it is a lower bound on the RL objective so that improving this bound guarantees improving upon agent performance.

MNM also optimizes a single objective for both model learning and policy optimization, except that the reward is log-transformed:

$$\max_{\hat{P},\pi} L_{\text{MNM}}(\hat{P},\pi) = \mathbb{E}_{Q(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t\left(\log R(s_t,a_t) + \log\frac{P(a_t|s_t)}{\pi(a_t|s_t)} + \log\frac{P(s_{t+1}|s_t,a_t)}{\hat{P}(s_{t+1}|s_t,a_t)}\right)\right]. \tag{29}$$

To evaluate MNM, the authors first showed that in a gridworld goal-reaching environment, MNM solves the task faster than VMBPO and Q-learning and it is robust to constraints on the dynamics model to only make low-resolution predictions. On robotic control tasks with contact dynamics and sparse reward (e.g., door-opening) in MuJoCo, Deepmind Control suite (DMC) (Tassa et al., 2018), Metaworld (Yu et al., 2020a), and ROBEL (Ahn et al., 2020) environments, MNM frequently outperformed MBPO and SAC by a large margin and it consistently performed well in all tasks. Furthermore, the authors found that MNM's model-based value estimates were stable and did not explode towards large values throughout the learning process, which suggest robustness to model-exploitation. Visualizing the learned dynamics, the authors also found the MNM dynamics model tends to generate transitions towards high value states, which provides evidence for the optimistic dynamics to speed up learning.

Ghugare et al. (2022) extended MNM to the setting of latent dynamics model with an algorithm called Latent Aligned Model (ALM) which is suitable for high-dimensional environments such as in visual RL. They designed the following prior and variational distributions:

$$P(\tau) = \mu(s_0)\prod_{t=0}^{\infty}P(s_{t+1}|s_t,a_t)P(a_t|z_t)E(z_t|s_t)\,, \tag{30}$$

$$Q(\tau) = \mu(s_0)E(z_0|s_0)\prod_{t=0}^{\infty}P(s_{t+1}|s_t,a_t)M(z_{t+1}|z_t,a_t)\pi(a_t|z_t)\,. \tag{31}$$

where $E(z|s)$ is an encoder, $M(z'|z,a)$ is a latent dynamics model, and $\pi(a|z)$ is a latent space policy.

Denoting $\hat{P} = \{E, M\}$, ALM's model and policy jointly optimize the following objective:

$$\max_{\hat{P},\pi} L_{\text{ALM}}(\hat{P},\pi) = \mathbb{E}_{Q(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t\left(\log R(s_t,a_t) + \log\frac{P(a_t|z_t)}{\pi(a_t|z_t)} + \log\frac{E(z_{t+1}|s_{t+1})}{M(z_{t+1}|z_t,a_t)}\right)\right]. \tag{32}$$

Interestingly, the third term in the augmented reward, which can be estimated using a classifier similar to MNM, is the information gain of the latent dynamics model, thus connecting this approach with prior work on intrinsic motivation and information-theoretic approaches in RL (Eysenbach et al., 2021; Rakelly et al., 2021; Sun et al., 2011; Millidge, 2020).

However, different from VMBPO and MNM's weighted-regression dynamics model learning objective, the ALM loss implies a "self-predictive" model learning process (Schwarzer et al., 2020), where the latent dynamics model is trained to predict the encoding of observations in the collected dataset and the encoder is trained to match the predictions of the latent dynamics (similar to Bootstrap Your Own Latent (Grill et al., 2020)). While theoretical understanding of self-predictive objectives is still lacking, Tang et al. (2023) proposed that it implicitly performs eigen-decomposition of the environment dynamics. Thus, the dynamics model learned by ALM is likely task-agnostic rather than task-aware as in VMBPO and MNM.

Empirically, the authors found that ALM significantly outperformed SAC and SAC-SVG (Amos et al., 2021), a MLE-based MBRL algorithm with latent dynamics, with 2e5 environment steps (1/5 of the usual RL training steps) in five MuJoCo environments. Using the Q value evaluation protocol from (Chen et al., 2021) and (Fujimoto et al., 2018), they found ALM value estimates to consistently have negative bias (underestimation). Similar to MNM, they also found that including the density ratio term in the objective is crucial.

Control-as-Inference Summary: The control-as-inference category is closely related to the distribution-correction category in that the variational distribution can be interpreted as the proposal distribution for an importance sampling estimator of the expected return. However, control-as-inference focuses more directly on return optimization as opposed to estimation as signified by the maximization of the likelihood of optimal trajectories.
The main advantage of the control-as-inference framework is that it provides a clear guidance to the derivation of model and policy objectives as long as a factorization of the prior and variational distributions are given. However, a major downside of current control-as-inference approaches is that these distributions are largely hand-designed, which is the key to the attractive properties in (32) but potentially a bottleneck for future objective design. Although automated design of posterior distributions leveraging graphical model factorization has been proposed for variational inference in probabilistic predictive models (Webb et al., 2018), extensions to the RL setting have not been considered or developed.

4.3 Value-Equivalence

A core use case of the learned dynamics model is that the agent can sample from it to augment the limited environment samples for estimating the expected return or value function. Thus, as long as the dynamics can lead to accurate value estimates, it does not need to model the environment at all. In other words, the model is free to discover any dynamics that are homomorphic to the true dynamics in terms of estimating value without wasting resources on irrelevant aspects of the environment. This class of approaches focuses almost entirely on model learning and performs standard model-based policy optimization. In this section, we first summarize how the literature formulates homomorphic dynamics and how identifying such dynamics can be formulated as a value-prediction problem. We then summarize how such homomorphism is related to the robustness properties of the learned dynamics.

4.3.1 Value-Prediction

The value-prediction approaches propose to directly predict states with accurate values rather than accurate features. This is desirable if the dynamics model has limited capacity modeling all aspects of the environment faithfully, for example when using a uni-modal distribution to predict multi-model transitions, but can generate states whose values are close to the ground truth future state values. The models don't necessarily have to predict the true expected return value which is unknown, and a major design decision in these approaches is what value function to predict.

As the first approach in this class, Farahmand et al. (2017) proposed to train the dynamics model such that it yields the same Bellman backup (i.e., (3)) as the ground truth model. They formulated this intuition as the following objective called Value-Aware Model Loss (VAML):

$$\min_{\hat{P}} L_{\text{VAML}}(\hat{P}) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \max_{V} \left| V(s') - \mathbb{E}_{s''\sim\hat{P}(\cdot|s,a)}[V(s'')] \right|^2 . \tag{33}$$

Since the true value function is not known, a robust formulation is used to optimize against the worst-case value function.

Empirically, the authors showed that VAML achieves smaller value-estimation error and higher return than MLE dynamics models in a finite MDP using a low-resolution dynamics model class.

To account for the fact that the optimizing policy π may be different from the behavior policy π_b, Voloshin et al. (2021) introduced density ratio correction into an objective similar to VAML. Furthermore, to account for unknown density ratio and value function, they propose to optimize against the worst-case for both in their Minimax Model Learning (MML) objective:

$$\min_{\hat{P}} L_{\text{MML}}(\hat{P}) = \max_{w,V} \left| \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[w(s,a) \left(V(s') - \mathbb{E}_{s''\sim\hat{P}(s'|s,a)}[V(s'')] \right) \right] \right| , \tag{34}$$

where $w(s,a) = \frac{d_P^{\pi}(s,a)}{d_P^{\pi_b}(s,a)}$ is the unknown density ratio between the optimizing policy and the behavior policy in the ground truth environment. They showed that this objective is a tighter bound on the policy evaluation error than the VAML objective.

Asadi et al. (2018) showed that the VAML loss function is equivalent to minimizing the Wasserstein distance between the learned and ground truth dynamics. As a result, the learned dynamics model tends to have attractive smoothness properties (e.g. being K-Lipschitz) and are less prone to compounding error and model exploitation.

However, the robust formulation of VAML poses a challenging optimization problem. As a response, Farahmand (2018) proposed to replace the worst-case value function with the most recent estimate in an objective called iterative VAML. In (Ayoub et al., 2020), the authors paired iterative VAML with an optimistic planning algorithm to derive a novel regret bound for the RL agent. Related to iterative VAML, MuZero (Schrittwieser et al., 2020; 2021) and Value Prediction Network (VPN) (Oh et al., 2017) replace the most recent value estimate with one bootstrapped from Monte Carlo Tree Search and multi-step look-ahead search, respectively (for a more nuanced discussion of VAML and MuZero, see (Voelcker et al., 2023)). A similar architecture and loss function was used in the Predictron model to evaluate the return of fixed deterministic policies (Silver et al., 2017a). MuZero has been shown to outperform prior state-of-the-art algorithms (e.g., AlphaZero; (Silver et al., 2017b)) in Go, Shogi, and Chess and large scale image-based online and offline RL settings (Schrittwieser et al., 2020; 2021).

Grimm et al. (2020) further extended the VAML approach by considering a set of policies Π and a set of arbitrary value functions $\mathcal{V}$ (i.e., not associated with any particular policies). They formulated the following objective based on the proposed Value-Equivalence (VE) principle:

$$\min_{\hat{P}} L_{\mathrm{VE}}(\hat{P};\Pi,\mathcal{V}) = \sum_{\pi\in\Pi}\sum_{V\in\mathcal{V}} \mathbb{E}_{(s)\sim\mathcal{D},a\sim\pi(\cdot|s)} \left\| \mathbb{E}_{s'\sim P(\cdot|s,a)}[V(s')] - \mathbb{E}_{s''\sim\hat{P}(\cdot|s,a)}[V(s'')] \right\| . \tag{35}$$

The primary difference between the VE loss and the VAML loss is that it is formulated using arbitrary policies and values, rather than just the data-collecting policies and its associated value estimates. The benefit of this is that as we increase the size of the set of policies and value functions considered, the space of value-equivalent models shrink before eventually reducing to the single ground truth model. Furthermore, when the model has limited capacity even at predicting values, one can trade off between the policy and value set considered and the desired value-equivalence loss (Grimm et al., 2021; 2022).

Empirically, the authors showed that VE significantly outperformed MLE on two tabular environments (Catch and Four Rooms) and Cartpole when rank constraint on the dynamics model was high and for a fixed rank-constrained model class, its performance improved with increasing size of the value function set.

Despite the simplicity and popularity of the VAML-family loss, Voelcker et al. (2022) suggested that it can cause undesirable optimization behavior when querying out-of-distribution value function estimates, especially when the learned model is not regularized to be close to the true environment. Using the inductive bias that the learned model should predict states similar to the true environment for next state samples s' in the dataset, they replaced the value function estimate in the VAML loss with its Taylor expansion around s': $\hat{V}(s)|_{s'} = V(s') + (\nabla_s V(s)|_{s'})^{\intercal}(s-s')$. Assuming deterministic dynamics and applying the Cauchy Schwartz inequality: $(\sum_{i=1}^{n} x_i)^2 \leq n\sum_{i=1}^{n} x_i^2$, the VAML loss reduces to the following Value-Gradient Weighted Model Loss (VaGradM):

$$\min_{\hat{P}} L_{\mathrm{VaGradM}}(\hat{P}) = \mathbb{E}_{(s,a,s')\sim\mathcal{D},s''\sim\hat{P}(\cdot|s,a)} \left[(s''-s')^{\intercal} diag(\nabla_s V(s)|_{s'})(s''-s')\right] . \tag{36}$$

When evaluated against VAML and MLE dynamics model training approaches with expressive dynamics (e.g., sufficiently large neural networks) in two MuJoCo environments, the authors found VaGradM to be robust to loss explosion when the value estimates were inaccurate in the initial training steps and converges to similar solutions to the MLE dynamics. However, VaGradM outperformed MLE when the dynamics model was constrained to fewer neural network layers and when distracting state dimensions were added.

The value-prediction approach can also be applied to policy-based RL methods, where instead of training the model to make accurate predictions of values, the model is trained to make accurate predictions of policy

gradients. Abachi (2020) proposed the (multi-step) Policy-Aware Model Loss (PAML) defined as follows:

$$\min_{\hat{P}} L_{\text{PAML}}(\hat{P}) = \left| \mathbb{E}_{\tau \sim P(\tau)} \left[\sum_{k=0}^{K} \gamma^k F(s_t, a_t) \right] - \mathbb{E}_{\tau \sim Q(\tau)} \left[\sum_{k=0}^{K} \gamma^k F(s_t, a_t) \right] \right|^2 , \tag{37}$$

where $Q(\tau) = \mu(s_0) \prod_{t=0}^{\infty} \hat{P}(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$, $F(s, a) = \nabla \log \pi(a|s) Q^*(s, a)$, and $Q^*(s, a)$ is a model-free value estimate.

In a linear-quadratic control setting, the author found PAML to learn more slowly and achieves lower asymptotic performance than MLE but performs substantially better when irrelevant state dimensions are added. However, this result did not transfer to the MuJoCo environments where PAML and MLE only performed on par regardless of whether there were irrelevant dimensions.

Value-Prediction Summary: Value-prediction approaches are similar to distribution correction and control-as-inference approaches in that it also tries to obtain more accurate value estimates, however, it focuses on estimating the Bellman operator as opposed to the return directly. Compared to two previous categories of approaches, value-prediction approaches have the benefit of requiring fewer moving parts, e.g., the importance weight estimator, and they do not by-default generate optimistic or pessimistic dynamics. However, by focusing on value-based RL algorithms, value-prediction approaches can be less flexible than the other classes of approaches.

4.3.2 Robust Control

Robust control approaches also aim to learn homomorphic dynamics to the ground truth. In contrast to value-prediction approaches, they highlight the pessimism in the learned dynamics model.

Modhe et al. (2020; 2021) suggested that the VAML and value-equivalence loss functions can be understood from the perspective of model advantage defined as:

$$A^{\pi}_{\hat{P}}(s, s') = \gamma \left[\mathbb{E}_{a \sim \pi(\cdot|s), s'' \sim \hat{P}(\cdot|s,a)} [V(s'')] - V(s') \right] , \tag{38}$$

where positive model advantage corresponds to optimistic dynamics and negative model advantage corresponds to pessimistic dynamics. However, by minimizing the norm of the value-prediction error, VAML and the VE optimize towards zero model advantage.

A more complete relationship between value-homomorphism and model advantage is depicted in (Vemula et al., 2023) where the authors proposed the following decomposition of the return gap between the optimizing policy π and the behavior policy π_b in terms of π's value in the learned model:

$$\begin{aligned}(1-\gamma)[J_P(\pi) - J_P(\pi_b)] = \quad & \underbrace{\mathbb{E}_{(s,a)\sim d^{\pi_b}_P} \left[V^{\pi}_{\hat{P}}(s) - Q^{\pi}_{\hat{P}}(s,a) \right]}_{\text{Model-based advantage under data distribution}} \\ & + \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{P}}} \left[\mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V^{\pi}_{\hat{P}}(s') \right] - \mathbb{E}_{s'' \sim \hat{P}(\cdot|s,a)} \left[V^{\pi}_{\hat{P}}(s'') \right] \right]}_{\text{Model dis-advantage under learner distribution}} \\ & + \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\pi_b}_P} \left[\mathbb{E}_{s'' \sim \hat{P}(\cdot|s,a)} \left[V^{\pi}_{\hat{P}}(s'') \right] - \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V^{\pi}_{\hat{P}}(s') \right] \right]}_{\text{Model advantage under data distribution}}\end{aligned} \tag{39}$$

This relationship suggests that the return gap between the two policies is more nuanced than just value-prediction error; it also involves model advantage under the data and learner distributions.

To improve upon the behavior policy, (39) suggests training the policy in the learned model starting from states visited by the behavior policy and simultaneously training the model to increase advantage under the data distribution and decrease advantage under the unknown learner distribution. To this end, they

proposed an algorithm called Lazy Model-based Policy Search (LAMPS) with the following model and policy objectives:

$$\begin{aligned}
&\max_{\hat{P}} L_{\text{LAMPS}}(\hat{P}) = \mathbb{E}_{(s,a)\sim D, s'\sim\hat{P}(\cdot|s,a)}\left[V^{\hat{\pi}}_{\hat{P}}(s')\right] - \mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{P}}, s'\sim\hat{P}(\cdot|s,a)}\left[V^{\pi}_{\hat{P}}(s')\right] \,, \\
&\max_{\pi} L_{\text{LAMPS}}(\pi) = \mathbb{E}_{s\sim D, a\sim\pi(\cdot|s)}[Q^{\pi}_{\hat{P}}(s,a)] \,.
\end{aligned} \tag{40}$$

where $d^{\pi}_{\hat{P}}$ is obtained by simulating the learner policy in the learned dynamics. By optimizing (40), the dynamics model will be optimistic under the data distribution, similar to control-as-inference approaches. However, the dynamics model will be pessimistic rather than just entropic outside the data distribution, resulting in a robust formulation.

Empirically, the authors found LAMPS to consistently achieve higher performance with fewer environment steps than MBPO across four MuJoCo environments.

In the context of offline RL, Uehara & Sun (2021) showed that policy solving the following optimization problem is competitive with any policy found on the offline dataset:

$$\max_{\pi} \min_{\hat{P}} J_{\hat{P}}(\pi), \quad \text{s.t. } \hat{P} \in \left\{P : \mathbb{E}_{(s,a)\sim\mathcal{D}} D_{TV}[P^{\text{MLE}}(\cdot|s,a)||P(\cdot|s,a)] \leq \epsilon\right\} \tag{41}$$

where P^{MLE} is a maximum likelihood estimate of the dynamics model.

Using this insight, Rigter et al. (2022) proposed the Robust Adversarial Model-based Offline Policy Optimization (RAMBO) algorithm with the following model and policy objectives:

$$\begin{aligned}
&\max_{\hat{P}} L_{\text{RAMBO}}(\hat{P}) = \lambda\mathbb{E}_{(s,a,s')\sim\mathcal{D}}[\log \hat{P}(s'|s,a)] - \mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{P}}, s'\sim\hat{P}(\cdot|s,a)}\left[V^{\pi}_{\hat{P}}(s')\right] \,, \\
&\max_{\pi} L_{\text{RAMBO}}(\pi) = \mathbb{E}_{d^{\pi}_{\hat{P}}(s,a)}[R(s,a)] \,.
\end{aligned} \tag{42}$$

where the hyperparameter λ controls adherence to the constraint. The RAMBO model objective can be interpreted as replacing the third row in (39) with the maximum likelihood objective, thus the learned dynamics model is only pessimistic. This is appropriate in the offline RL setting since the agent is not allowed to interact with the environment to explore and correct for optimistic mistakes.

Empirically, the authors showed that RAMBO achieves the best overall performance amongst other model-based and model-free baselines in four MuJoCo tasks with varying dataset qualities and it significantly outperforms other model-based baselines in the AntMaze environment which has more challenging contact dynamics. Compared with COMBO (Yu et al., 2021), a SOTA model-based offline RL algorithm, RAMBO learns a smoother dynamics model, potentially making it less prone to local optima.

Robust Control Summary: Robust control approaches reveal a hidden insight behind value-prediction approaches: in the process of finding homomorphic MDPs, the dynamics model is actually becoming optimistic on some state-action pairs and pessimistic on others depending on its visitation. The inherent pessimism makes robust control approaches especially suited for addressing distribution-shift, such as in offline RL.

4.4 Differentiable Planning

Instead of explicitly defining model learning objectives, differentiable planning approaches embed the policy (or trajectory) optimization process with respect to the learned model as a differentiable program and update the model with respect to a higher level objective, such as maximizing return in the true environment or the standard Bellman error loss using environment samples (Mnih et al., 2013). These approaches typically take on a bi-level optimization format where optimality with respect to the learned model is defined as a constraint.

Farquhar et al. (2018) first recognized that for discrete actions and deterministic dynamics, the multi-step look-ahead search in VPN (Oh et al., 2017) can be interpreted as an neural network layer, which can be used

to process state inputs before outputting the final value prediction. They proposed the TreeQN architecture which can be formulated as the following bi-level optimization problem:

$$\begin{aligned} \min_{\hat{P},Q} \quad & L_{\text{TreeQN}} = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left(R(s,a) + \gamma V(s') - Q(s,a)\right)^2 \\ \text{s.t.} \quad & V(s) = \arg\max_{a_{0:K}} \mathbb{E}_{\hat{P}} \left[\sum_{k=0}^{K-1} \gamma^k R(s_k, a_k) + \gamma^K V(s_K) | s_0 = s\right]. \end{aligned} \tag{43}$$

In a box-pushing environment, the authors showed that TreeQN significantly outperformed DQN (Mnih et al., 2013) and its performance improved when increasing the look-ahead depth from 1 to 3 steps. In the Atari environments, TreeQN outperformed DQN in all except 1 environment and consistently achieved higher performance with fewer environment steps. However, the effect of look-ahead depth in Atari were not observable.

Nikishin et al. (2022) proposed a similar bi-level optimization approach called Optimal Model Design (OMD). However, instead of using a multi-step look-ahead search, the constraint was defined using the first-order optimality condition for Bellman error minimization with respect to the learned model. The OMD model and policy objectives are defined as follows:

$$\begin{aligned} \min_{\hat{P},Q} \quad & L_{\text{OMD}} = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left(R(s,a) + \gamma V(s') - Q(s,a)\right)^2 \\ \text{s.t.} \quad & \nabla_{\hat{P}} \mathbb{E}_{(s,a)\sim\mathcal{D}, s'\sim\hat{P}(\cdot|s,a)} \left(R(s,a) + \gamma V(s') - Q(s,a)\right)^2 = 0 \end{aligned} \tag{44}$$

where the soft-value function $V(s) := \log \sum_a \exp(Q(s,a))$ was used to make the constraint differentiable.

A novelty of the OMD approach is the use of the implicit function theorem to differentiate through the constraint. This method was later extended in (Bansal et al., 2023) where only a metric on the model loss was included in the upper objective to automatically weight task-relevant features while the standard prediction objective under the optimized metric was still used for model learning and formulated as an additional constraint. Sharma et al. (2023) used a similar implicit differentiation approach in a setting where the reward function is sampled from a distribution. Wang et al. (2021) used a sample-based approximation of implicit gradients in the setting of predicting missing parameters in an MDP from logged trajectories and then perform offline RL.

Empirically, the authors showed that OMD outperformed MLE in Cartpole and one MuJoCo environment when the dynamics model hidden dimensions or parameter norm were constrained and when distracting state dimensions were added.

In contrast to the above value-based upper objectives, Amos & Yarats (2020) proposed to re-parameterize the policy by embedding the model in a differentiable Cross Entropy Method (DCEM) trajectory optimizer. Each action outputted by the DCEM policy is computed by running the DCEM trajectory optimization algorithm for a finite number of iterations. Then, the dynamics model parameters are updated with respect to the upper level objective of the expected return of the policy in the true environment. The DCEM model and policy objectives are defined as follows:

$$\begin{aligned} \max_{\hat{P}} \quad & L_{\text{DCEM}}(\hat{P}) = \mathbb{E}_{d^{\pi}_{\hat{P}}(s,a)}[R(s,a)] \\ \text{s.t.} \quad & \pi(a|s;\hat{P}) = \arg\max_{a_{0:K}} \mathbb{E}_{\hat{P}} \left[\sum_{k=0}^{K} R(s_k, a_k) | s_0 = s, a_0 = a\right] \end{aligned} \tag{45}$$

The upper level objective is optimized using the Proximal Policy Optimization algorithm (Schulman et al., 2017b).

Empirically, the authors showed that DCEM matched the performance of Dreamer (Hafner et al., 2019), a popular MBRL algorithm, on two MuJoCo tasks with an order-of-magnitude fewer environment samples.

Prior to the above deep learning based differentiable planners, Joseph et al. (2013) and Bansal et al. (2017) explored similar bi-level optimization formulations of model learning where the model parameters are updated

using finite-difference gradient approximation and Bayesian optimization, respectively. These methods were applied to simple and low-dimensional problems and they are difficult to scale to high-dimensional settings with large dynamics model parameterizations.

Recently, Gehring et al. (2021) theoretically studied differentiable planners in a simplified reward prediction setting and found that their learning dynamics can be understood as pre-conditioned gradient descent, which can significantly increase convergence speed depending on the initialization of the initial value estimate.

Despite limited applications in RL, differentiable planners are a popular approach for inverse RL and imitation learning (Tamar et al., 2016; Okada et al., 2017; Srinivas et al., 2018; Karkus et al., 2017; Amos et al., 2018). Leveraging the relationship between value-equivalence and robust control (i.e., (39)), Wei et al. (2023) showed that favorable properties of differentiable inverse planners can be attributed to learning pessimistic dynamics and robust policies.

Differentiable Planning Summary: Differentiable planning represents the most direct approach to solving objective mismatch by embedding a minimal set of standard MBRL components (e.g., only a model and a policy) in a differentiable program so that all components optimize the same objective. This class of approaches bypasses return estimation and focuses directly on return optimization. It also involves the least amount of human intervention in the objective design process.

5 Discussion

This paper reviewed solutions to the objective mismatch problem in MBRL and classified the approaches into a taxonomy based on their structure and intuition. The taxonomy highlights that these approaches have influences on key components of MBRL which we discuss below:

- MBRL objective design and the search for value optimization-equivalence in both the model and the policy.
- Important agent properties, such as ground truth model identifiability, agent exploration behavior, and model optimism and pessimism.
- Optimization approaches and how to extract maximum information from novel objectives.
- Downstream applications and re-using dynamics models.
- Benchmarking and improving rigor in MBRL.

5.1 Decision-Aware Objective Design

The proposed taxonomy suggests a single principle to decision-aware model and policy objective design: *value optimization-equivalence*, where both the model and the policy should be trained to optimize the expected return in the real environment. Under this principle, we can identify a split among the reviewed approaches similar to the policy-based vs. value-based and model-free vs. model-based paradigms in RL and whether these approaches are designed to explicitly handle errors occurring at different stages of the learning process.

Error-awareness: Error-aware approaches in the distribution correction and control-as-inference categories achieve value optimization-equivalence by explicitly modeling the errors of simulated trajectories using distribution correction and adapting the model learning and policy optimization objective. Methods that only address model-shift, e.g., by redesigning policy objectives, can be more permissive towards the specific model-training method. However, authors have also identified model-training objectives with better optimization behavior from the policy evolution, policy-gradient matching, variance reduction, or control-as-inference perspectives.

On the other hand, error-agnostic approaches achieve value-optimization equivalence by constructing equivalent classes of MDP dynamics, either explicitly as in value-equivalence or implicitly as in differentiable

Table 1: Comparisons of architecture and implementation of the core reviewed decision-aware MBRL algorithms. For each algorithm, we list its objective mismatch solution category (*DC=distribution correction, MS=model shift, PS=policy shift, CAI=control-as-inference, VE=value equivalence, VP=value prediction, RC=robust control, DP=differentiable planning*), dynamics model type, whether ensembling of dynamics was used, whether the dynamics model was trained to make multi-step predictions, the policy optimization algorithm, and other agent components.

Algorithm	Category	Model type	Ensemble	Multi-step	Policy algo.	Other components
DAM (Haghgoo et al., 2021)	DC-MS	Mixture density network	No	No	Shooting	Classifier
TOM (Ma et al., 2023)	DC-PS	Mixture density network	Yes	No	SAC	Classifier, model $\tilde{Q}$
AMPL (Yang et al., 2022)	DC-PS	Gaussian	Yes	No	TD3	Classifier, IW estimator
GAMPS (D'Oro et al., 2020)	DC-PS	Linear Gaussian	No	No	PG	-
VMBPO (Chow et al., 2020)	CAI	Gaussian	Yes	No	SAC	Classifier
MNM (Eysenbach et al., 2022)	CAI	Gaussian	Yes	No	SAC	Classifier
ALM (Ghugare et al., 2022)	CAI	Latent Gaussian	No	Yes	DDPG	Classifier
VAML (Farahmand et al., 2017)	VE-VP	Tabular	No	No	VI	Adversarial V
MML (Voloshin et al., 2021)	VE-VP	Gaussian	No	No	Discrete SAC	Adversarial V & IW
MuZero (Schrittwieser et al., 2020)	VE-VP	Latent deterministic	No	Yes	DQN, MCTS	-
VPN (Oh et al., 2017)	VE-VP	Latent deterministic	No	Yes	DQN, TD search	-
VE (Grimm et al., 2020)	VE-VP	Deterministic	No	No	DQN	-
VaGradM (Voelcker et al., 2022)	VE-VP	Deterministic	Yes	No	SAC	-
PAML (Abachi, 2020)	VE-VP	Deterministic	No	Yes	DDPG	-
LAMPS (Vemula et al., 2023)	VE-RC	Gaussian	Yes	No	SAC	-
RAMBO (Rigter et al., 2022)	VE-RC	Gaussian	Yes	No	SAC	-
TreeQN (Farquhar et al., 2018)	DP	Deterministic	No	No	DQN	-
OMD (Nikishin et al., 2022)	DP	Deterministic	No	No	Discrete SAC	-
DCEM (Amos & Yarats, 2020)	DP	Latent deterministic	No	No	CEM	-

Table 2: Comparisons of evaluation and experiments of the core reviewed decision-aware MBRL algorithms. For each algorithm, we list its objective mismatch solution category (*DC=distribution correction, MS=model shift, PS=policy shift, CAI=control-as-inference, VE=value equivalence, VP=value prediction, RC=robust control, DP=differentiable planning*), main evaluation environments, main baseline algorithms, the types of model misspecification evaluated, and the metric used for evaluating model exploitation (policy gradient or Q estimation accuracy).

Algorithm	Category	Environments	Baselines	Misspecification	Exploitation
DAM (Haghgoo et al., 2021)	DC-MS	Driving	MLE	Multimodal	-
TOM (Ma et al., 2023)	DC-PS	MuJoCo	MBPO, PDML, VaGradM	-	-
AMPL (Yang et al., 2022)	DC-PS	Maze2D, MuJoCo, Adroit	COMBO, CQL	-	-
GAMPS (D'Oro et al., 2020)	DC-PS	MuJoCo	PG	Remove features	PG
VMBPO (Chow et al., 2020)	CAI	MuJoCo	MBPO, SAC	-	-
MNM (Eysenbach et al., 2022)	CAI	MuJoCo, DMC, Metaworld, ROBEL	MBPO, SAC	Low-res.	Q
ALM (Ghugare et al., 2022)	CAI	MuJoCo	MBPO, SAC-SVG	-	Q
VAML (Farahmand et al., 2017)	VE-VP	Tabular	MLE	Low-res.	Q
MML (Voloshin et al., 2021)	VE-VP	LQR, Cartpole	MLE, VAML	-	Q
MuZero (Schrittwieser et al., 2020)	VE-VP	Atari, Go, Chess, Shogi	AlphaZero	-	-
VPN (Oh et al., 2017)	VE-VP	Atari	DQN, MLE	-	-
VE (Grimm et al., 2020)	VE-VP	Four rooms, Catch, Cartpole	MLE	Rank	-
VaGradM (Voelcker et al., 2022)	VE-VP	MuJoCo	MBPO, VAML	NN size, distractor	-
PAML (Abachi, 2020)	VE-VP	MuJoCo	MLE, DDPG	Distractor	-
LAMPS (Vemula et al., 2023)	VE-RC	MuJoCo	MBPO	-	-
RAMBO (Rigter et al., 2022)	VE-RC	MuJoCo, AntMaze	COMBO, CQL	-	Q
TreeQN (Farquhar et al., 2018)	DP	Atari	DQN, A2C	-	-
OMD (Nikishin et al., 2022)	DP	MuJoCo, Cartpole	MLE	NN size	-
DCEM (Amos & Yarats, 2020)	DP	DMC	Dreamer	-	-

planning, such that evaluating policies in the learned dynamics is close to that of the true dynamics. These approaches simplify the training pipeline as they do not alter the standard model-based value estimation procedure, for example by adding density-ratio estimators and auxiliary model-based reward bonus as in DAM and MNM.

However, many approaches in both the error-aware and error-agnostic categories still do not explicitly control for policy-shift (i.e., limiting policy divergence in (6)) or introduce policy-shift to new agent components. In error-aware approaches, the value optimization-equivalence property may be hindered if the density-ratio estimator trained on past data cannot generalize to new policies. Similarly, in error-agnostic approaches, minimizing the value-prediction loss on past data may not ensure small loss on new data. Instead, policy-shift is implicitly addressed by taking small optimization steps and relying on fast alternation between model and policy updates (Ma et al., 2023), optimizing against the worst-case density-ratio (Voloshin et al., 2021), against the worst-case dynamics (Rigter et al., 2022), or introducing reward bonus (Haghgoo et al., 2021; Eysenbach et al., 2022; Yang et al., 2022).

Value-awareness: In RL, policy-based and value-based approaches are distinguished by whether a value function is a necessary component of the algorithms, and policy-based approaches can operate on Monte Carlo return estimators rather than explicit value function estimators. Using this analogy, we can classify distribution correction and differentiable planning as policy-based and control-as-inference and value-equivalence as value-based. Similar to policy optimization in RL, value-based approaches for model learning are tied to the quality of the value function estimator, which can require special care either in objective design (e.g., identifying the extrapolation errors of value function estimates (Voelcker et al., 2022; Farahmand et al., 2017; Voloshin et al., 2021)) or implementation (e.g., making sure the value function estimator is sufficiently accurate before applying to model learning (Eysenbach et al., 2022)).

Model-awareness: The value optimization-equivalence principle also implies a hybridization of model-free and model-based RL approaches, which is manifested in most reviewed works. The traditional view of model-based approaches is to build an as accurate as possible model of the environment. The value optimization-equivalence principle suggests to instead stay as close as possible to model-free RL in the sense that sampling from the model or evaluating samples from the model is close to having samples drawn from the true environment. This is intuitive because model-free RL evaluates returns on true samples; this is as if we were running MBRL with privileged knowledge of the true dynamics. However, there are still several advantages of MBRL that are beyond just having better environment sample complexity. As Gehring et al. (2021) suggested, value optimization-equivalent MBRL likely enjoys optimization advantages due to the relationship with accelerated gradient descent methods. Value optimization-equivalent MBRL may also enjoy exploration and safety advantages depending on whether and in what part of the state-action space the learned dynamics is optimistic or pessimistic.

5.2 Properties of Decision-Aware Agents

Decision-aware MBRL methods can mirror the enactive view of cognitive science (Di Paolo & Thompson, 2014), where the role of perception is not to build a true model of the environment but to serve as an interface between the agent and the environment to enable flexible and adaptive behavior. Under this view, the learned model is free to deviate from the environment. It is thus useful to understand how learned models deviate from the true model, how agents behave during the learning process, and implications for downstream applications.

The fullest picture is depicted by Vemula et al. (2023) who showed in an exact decomposition of the return gap that the learned model is optimistic in the in-distribution region of the state-action space and pessimistic in the out-of-distribution region. The adversarial model learning approach by Rigter et al. (2022) adopts the pessimistic perspective, but instead learns an objective (accurate) model in the in-distribution region. Agents trained with this objective will likely be more conservative in their exploration behavior and less prone to model-exploitation. In contrast, control-as-inference approaches learn an optimistic model of the dynamics in the in-distribution region, and they do not explicitly model out-of-distribution behavior in the model-learning objective. Instead, control-as-inference learns conservative behavior in the policy-optimization process by encouraging agents to follow well-predicted states and only deviate when additional information about the

dynamics model can be gained. These agents will likely exhibit more exploratory behavior than robust control agents in a more structured manner. For example, Eysenbach et al. (2022) found that their agent tends to make the goal positions appear closer than they actually are in model rollouts so that the goals are easier to reach, which is beneficial in sparse reward settings to accrue more feedback for the policy. Control-as-inference agents also represent an interesting point of contact with recent information-theoretic approaches to agent objective design in both task-driven and unsupervised RL (Hafner et al., 2020; Rhinehart et al., 2021).

A substantial question for value optimization-equivalent agents is whether the true environment model can be identified. The findings of Nikishin et al. (2022) suggest that the true environment model is not identifiable given that two different dynamics models can lead to the same optimal policy. This equivalence is related to the research on state-abstraction in MDPs where functionally similar states are aggregated (in the current case ignored) in order to avoid modeling task-irrelevant state features (Li et al., 2006). One way to improve identifiability is to add an environment or reward prediction loss as in (Schrittwieser et al., 2020; 2021), however, this introduces a trade-off between value-equivalent vs. accurate models which has to be specified by the modeler. Nikishin et al. (2022) suggested that unidentifiability is likely a blessing rather than a curse because there are likely value-equivalent models that are easier to learn than the true model. However, Wei et al. (2023) showed in an inverse RL setting with differentiable planners that favorable robustness properties of value-equivalent models may succumb to overly high inaccuracy. They further suggested regularizing the model with state-prediction accuracy. As noted by several authors, the set of value-equivalent models reduces with increasing number of policies and values considered (Grimm et al., 2020; Voloshin et al., 2021). This suggests that identifiability is possible in a multi-task or meta-learning setting and an experimental validation is provided in (Berke et al., 2022) from a cognitive science perspective. However, currently all works on decision-aware MBRL has focused on the single-task setting.

5.3 Optimization Approaches

In order to gain the full benefit from novel objective formulations designed to solve objective mismatch, different optimization techniques may be needed. We highlight two optimization approaches from the survey. The majority of reviewed works use manually designed objectives for model learning and policy optimization mostly by bounding the true return and leveraging existing optimization techniques such as policy-gradient and adversarial training. However, certain properties of decision-aware agents can be lost in manual formulation and optimization. For example, optimistic or pessimistic behavior characteristic of decision-aware agents is lost in models learned using value-prediction. In contrast to manual design of component objectives, differentiable planning approaches use a single objective of the true return and offload the complexity of optimization to differentiable programming. These approaches mostly take on a bilevel-optimization format, where an update of both the model and the policy need to take into account the optimal policy with respect to the current model. To this end, earlier differentiable planners such as (Farquhar et al., 2018; Amos & Yarats, 2020) relied on finite-horizon or truncated-horizon policy objectives where the gradients can be computed using backpropagation. More recent approaches such as (Nikishin et al., 2022; Bansal et al., 2023) have explored alternative optimization techniques such as implicit differentiation which can more precisely compute the gradient of infinite-horizon policies. However, the authors have also commented on the added complexity and approximation to these approaches. While manually designed optimization is advantageous for understanding the properties of decision-aware agents, more efficient end-to-end optimization approaches may be desirable for scaling and broadening application domains.

5.4 Downstream Applications

An important use case for MBRL is transfer learning, where the dynamics model learned in a source task can be used as an environment simulator for a target task to reduce or eliminate the number of environment samples (Taylor & Stone, 2009; Zhu et al., 2023). For traditional environment prediction-based model learning approaches, transferring would simply require the model to generalize in state-action space covered by the target task. However, transfer learning presents a larger challenge for value optimization-equivalent agents because of model unidentifiability and the fact that decision-aware models are usually biased. In this setting, it is not clear whether model bias, especially the optimistic bias, which aids optimization in the

source task is also helpful for the target task. However, given that model-identifiability can be improved with multi-task training, decision-aware agents may be especially suited for tasks that require continuous model fine-tuning or adaptation such as in some instances of meta RL (Nagabandi et al., 2018b;a) and lifelong learning (Khetarpal et al., 2022).

Another important utility of MBRL is to enhance the transparency and explainability of RL agents through the separation of model and policy, where designers can introspect the learned model to understand agent behavior and potentially correct agent failures. An important aspect of transparency is that the agent designer can easily comprehend and identify sub-optimalities in the model and make precise editing decisions (Räuker et al., 2023). The biases and unidentifiability in decision-aware agents introduce significant challenges for model comprehension since the true environment is no longer the ground truth or the optimization target and it may not be appropriate to correct the model towards the ground truth only on some identified states but not others since it may alter the learned value-equivalence.

These application considerations suggest a need to better understand the properties of decision-aware MBRL agents, such as MDP homomorphism, in order to reap their benefits without losing that of traditional MBRL.

5.5 Evaluations and Benchmarks

While resolving objective mismatch focuses on designing novel agent objectives or training procedures, the qualities of the objectives should be measured based on agent behavior. Since the ultimate goal of decision-aware agents is to achieve high returns, the final performance and learning speed (i.e., the number of environment steps to reach a performance threshold) are the primary evaluation metrics which have been used by most reviewed decision-aware MBRL work. However, more fine-grained evaluation of decision-aware MBRL should probe its advantages over traditional MBRL and model-free RL, most importantly, robustness to model misspecification and model-exploitation. Many reviewed works such as (D'Oro et al., 2020; Eysenbach et al., 2022; Farahmand et al., 2017; Voelcker et al., 2022; Abachi, 2020; Nikishin et al., 2022; Grimm et al., 2020) compared agent performance under different levels of model misspecification by varying the capacity of the dynamics models, removing state features, or adding distracting state features (see Table 2). Evaluation protocols of model-exploitation have also been developed in recent years by measuring value-estimation bias (Chen et al., 2021; Fujimoto et al., 2018). We recommend future work on decision-aware MBRL to include these experiments.

Beyond model misspecification and exploitation, some authors also probed more specific agent properties (e.g., exploration behavior). For example, Eysenbach et al. (2022) and Rigter et al. (2022) assessed the optimism and smoothness, respectively, of their learned dynamics models through visualizations in selected environments. Since these properties do not necessarily apply to all decision-aware MBRL approaches, different works performed different evaluations in different environments, which makes comparison of approaches incomplete if not challenging. Thus, as our understanding of the properties and utilities of decision-aware agents continues to develop and mature, we may benefit from documenting the agent properties that each environment is designed to test and developing behavior suites (Osband et al., 2019) for decision-aware agents.

The consistency between theory and implementation in decision-aware MBRL also warrants additional attention. Several works have remarked on the theory-implementation gap (Eysenbach et al., 2022; Modhe et al., 2021; Lovatto et al., 2020), which are mostly due to additional moving components and optimization challenges (e.g., requirements on value estimation accuracy as a result of including value estimates in model training objectives). Thus, evaluations of decision-aware MBRL should focus on not only the final performance but also implementation easiness, training stability, debugging tools (Lambert, 2021), sharing trained models (Pineda et al., 2021), and other optimization failure modes.

6 Conclusion

While resolving objective mismatch promises to boost the capability of MBRL agents, how to design aligned objectives for different agent components remains an open question. To this end, we found that all current efforts to address the objective mismatch problem can be understood along the lines of 4 major categories:

distribution correction, control-as-inference, value-equivalence, and differentiable planning. These efforts point to a single principle for designing decision-aware objectives: *value optimization-equivalence.* Under this principle, both the dynamics model and the policy should be trained to optimize the expected return, effectively achieving a hybridization of model-free and model-based RL. We recommend future work to continue to enhance our understanding of decision-aware agents, identify their practical utilities, and design appropriate evaluation suites to fully harvest their benefits.

Broader Impact

This paper synthesizes theoretical and empirical results for solving objective mismatch towards building more capable MBRL agents. RL and automated decision-making system can have important ramifications, especially when directly interfacing with humans. A major subset of these ramifications stems from misspecified reward and training environments. While we have focused on correctly specified reward and training environments, we remark that identifying and debugging these misspecifications in decision-aware MBRL agents is likely harder than traditional RL agents. We believe developing better theoretical understanding and empirical testing suites are a first step towards the transparency required to mitigate harms from decision-aware MBRL agents.

References

Romina Abachi. *Policy-aware model learning for policy gradient methods.* University of Toronto (Canada), 2020.

David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1639–1650. PMLR, 2020.

Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pp. 1300–1313. PMLR, 2020.

Brandon Amos and Denis Yarats. The differentiable cross-entropy method. In *International Conference on Machine Learning*, pp. 291–302. PMLR, 2020.

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.

Brandon Amos, Samuel Stanton, Denis Yarats, and Andrew Gordon Wilson. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, pp. 6–20. PMLR, 2021.

Kavosh Asadi, Evan Cater, Dipendra Misra, and Michael L Littman. Equivalence between wasserstein and value-aware loss for model-based reinforcement learning. *arXiv preprint arXiv:1806.01265*, 2018.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pp. 463–474. PMLR, 2020.

Dishank Bansal, Ricky Chen, Mustafa Mukadam, and Brandon Amos. Taskmet: Task-driven metric learning for model learning. In *Differentiable Almost Everything Workshop*, 2023.

Somil Bansal, Roberto Calandra, Ted Xiao, Sergey Levine, and Claire J Tomlin. Goal-driven dynamics learning via bayesian optimization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5168–5173. IEEE, 2017.

Marlene D Berke, Robert Walter-Terrill, Julian Jara-Ettinger, and Brian J Scholl. Flexible goals require that inflexible perceptual systems produce veridical representations: Implications for realism as revealed by evolutionary simulations. *Cognitive Science*, 46(10):e13195, 2022.

Veronica Chelu, Doina Precup, and Hado P van Hasselt. Forethought and hindsight in credit assignment. *Advances in Neural Information Processing Systems*, 33:2270–2281, 2020.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

Yinlam Chow, Brandon Cui, MoonKyung Ryu, and Mohammad Ghavamzadeh. Variational model-based policy optimization. *arXiv preprint arXiv:2006.05443*, 2020.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Ezequiel Di Paolo and Evan Thompson. The enactive approach. *The Routledge handbook of embodied cognition*, pp. 68–78, 2014.

Pierluca D'Oro, Alberto Maria Metelli, Andrea Tirinzoni, Matteo Papini, and Marcello Restelli. Gradient-aware model-based policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3801–3808, 2020.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Robust predictable control. *Advances in Neural Information Processing Systems*, 34:27813–27825, 2021.

Benjamin Eysenbach, Alexander Khazatsky, Sergey Levine, and Russ R Salakhutdinov. Mismatched no more: Joint model-policy optimization for model-based rl. *Advances in Neural Information Processing Systems*, 35:23230–23243, 2022.

Amir-massoud Farahmand. Iterative value-aware model learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pp. 1486–1494. PMLR, 2017.

Gregory Farquhar, Tim Rocktäschel, Maximilian Igl, and Shimon Whiteson. Treeqn and atreec: Differentiable tree-structured models for deep reinforcement learning. *arXiv preprint arXiv:1710.11417*, 2018.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Clement Gehring, Kenji Kawaguchi, Jiaoyang Huang, and Leslie Kaelbling. Understanding end-to-end model-based reinforcement learning methods as implicit parameterization. *Advances in Neural Information Processing Systems*, 34:703–714, 2021.

Raj Ghugare, Homanga Bharadhwaj, Benjamin Eysenbach, Sergey Levine, and Ruslan Salakhutdinov. Simplifying model-based rl: Learning representations, latent-space models, and policies with one objective. *arXiv preprint arXiv:2209.08466*, 2022.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, pp. 589–590. MIT press, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:5541–5552, 2020.

Christopher Grimm, André Barreto, Greg Farquhar, David Silver, and Satinder Singh. Proper value equivalence. *Advances in Neural Information Processing Systems*, 34:7773–7786, 2021.

Christopher Grimm, Andre Barreto, and Satinder Singh. Approximate value equivalence. *Advances in Neural Information Processing Systems*, 35:33029–33040, 2022.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Pedro A Ortega, Jimmy Ba, Thomas Parr, Karl Friston, and Nicolas Heess. Action and perception as divergence minimization. *arXiv preprint arXiv:2009.01791*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Behzad Haghgoo, Allan Zhou, Archit Sharma, and Chelsea Finn. Discriminator augmented model-based reinforcement learning. *arXiv preprint arXiv:2103.12999*, 2021.

Toru Hishinuma and Kei Senda. Weighted model estimation for offline model-based reinforcement learning. *Advances in neural information processing systems*, 34:17789–17800, 2021.

Taher Jafferjee, Ehsan Imani, Erin Talvitie, Martha White, and Micheal Bowling. Hallucinating value: A pitfall of dyna-style planning with imperfect environment models. *arXiv preprint arXiv:2006.04363*, 2020.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Michael Janner, Igor Mordatch, and Sergey Levine. gamma-models: Generative temporal difference learning for infinite-horizon prediction. *Advances in Neural Information Processing Systems*, 33:1724–1735, 2020.

Joshua Joseph, Alborz Geramifard, John W Roberts, Jonathan P How, and Nicholas Roy. Reinforcement learning with misspecified model classes. In *2013 IEEE International Conference on Robotics and Automation*, pp. 939–946. IEEE, 2013.

Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, 30, 2017.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49:209–232, 2002.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Didolkar, Dipendra Misra, Dylan Foster, Lekan Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of controllable latent states with multi-step inverse models. *arXiv preprint arXiv:2207.08229*, 2022.

Nathan Lambert. Debugging model-based reinforcement learning systems. 2021.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

Nathan Lambert, Albert Wilcox, Howard Zhang, Kristofer SJ Pister, and Roberto Calandra. Learning accurate long-term dynamics for model-based reinforcement learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2880–2887. IEEE, 2021.

Nathan Lambert, Kristofer Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *arXiv preprint arXiv:2203.09637*, 2022.

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. *arXiv preprint arXiv:1909.01506*, 2019.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *AI&M*, 2006.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Ângelo G Lovatto, Thiago P Bueno, Denis D Mauá, and Leliane N Barros. Decision-aware model learning for actor-critic methods: when theory does not meet practice. 2020.

Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *arXiv preprint arXiv:2206.09328*, 2022.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Yecheng Jason Ma, Kausik Sivakumar, Jason Yan, Osbert Bastani, and Dinesh Jayaraman. Learning policy-aware models for model-based reinforcement learning via transition occupancy matching. In *Learning for Dynamics and Control Conference*, pp. 259–271. PMLR, 2023.

Rowan McAllister, Blake Wulfe, Jean Mercat, Logan Ellis, Sergey Levine, and Adrien Gaidon. Control-aware prediction objectives for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 01–08. IEEE, 2022.

Beren Millidge. Deep active inference as variational policy gradients. *Journal of Mathematical Psychology*, 96:102348, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Nirbhay Modhe, Harish K Kamath, Dhruv Batra, and Ashwin Kalyan. Bridging worlds in reinforcement learning with model-advantage. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

Nirbhay Modhe, Harish Kamath, Dhruv Batra, and Ashwin Kalyan. Model-advantage and value-aware models for model-based reinforcement learning: Bridging the gap in theory and practice. *arXiv preprint arXiv:2106.14080*, 2021.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

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. *arXiv preprint arXiv:1803.11347*, 2018a.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018b.

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7886–7894, 2022.

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. *Advances in neural information processing systems*, 30, 2017.

Masashi Okada, Luca Rigazio, and Takenobu Aoshima. Path integral networks: End-to-end differentiable optimal control. *arXiv preprint arXiv:1706.09597*, 2017.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvari, Satinder Singh, et al. Behaviour suite for reinforcement learning. *arXiv preprint arXiv:1908.03568*, 2019.

Luis Pineda, Brandon Amos, Amy Zhang, Nathan O Lambert, and Roberto Calandra. Mbrl-lib: A modular library for model-based reinforcement learning. *arXiv preprint arXiv:2104.10159*, 2021.

Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. Which mutual-information representation learning objectives are sufficient for control? *Advances in Neural Information Processing Systems*, 34:26345–26357, 2021.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 464–483. IEEE, 2023.

Nicholas Rhinehart, Jenny Wang, Glen Berseth, John Co-Reyes, Danijar Hafner, Chelsea Finn, and Sergey Levine. Information is power: intrinsic control via information capture. *Advances in Neural Information Processing Systems*, 34:10745–10758, 2021.

Spencer M Richards, Jean-Jacques Slotine, Navid Azizan, and Marco Pavone. Learning control-oriented dynamical structure from data. *arXiv preprint arXiv:2302.02529*, 2023.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *arXiv preprint arXiv:2204.12581*, 2022.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34:27580–27591, 2021.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Abhishek Sharma, Sonali Parbhoo, Omer Gottesman, and Finale Doshi-Velez. Robust decision-focused learning for reward transfer. *arXiv preprint arXiv:2304.03365*, 2023.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Hado Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. The predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, pp. 3191–3199. PMLR, 2017a.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017b.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

Sumeet Singh, Spencer M Richards, Vikas Sindhwani, Jean-Jacques E Slotine, and Marco Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. *The International Journal of Robotics Research*, 40(10-11):1123–1150, 2021.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pp. 4732–4741. PMLR, 2018.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*, pp. 41–51. Springer, 2011.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *Advances in neural information processing systems*, 29, 2016.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding self-predictive learning for reinforcement learning. In *International Conference on Machine Learning*, pp. 33632–33656. PMLR, 2023.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark Neerincx. Contrastive explanations for reinforcement learning in terms of expected consequences. *arXiv preprint arXiv:1807.08706*, 2018.

Anirudh Vemula, Yuda Song, Aarti Singh, J Andrew Bagnell, and Sanjiban Choudhury. The virtues of laziness in model-based rl: A unified objective and algorithms. *arXiv preprint arXiv:2303.00694*, 2023.

Claas Voelcker, Victor Liao, Animesh Garg, and Amir-massoud Farahmand. Value gradient weighted model-based reinforcement learning. *arXiv preprint arXiv:2204.01464*, 2022.

Claas Voelcker, Arash Ahmadian, Romina Abachi, Igor Gilitschenski, and Amir-massoud Farahmand. λ-ac: Learning latent decision-aware models for reinforcement learning in continuous state-spaces. *arXiv preprint arXiv:2306.17366*, 2023.

Cameron Voloshin, Nan Jiang, and Yisong Yue. Minimax model learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1612–1620. PMLR, 2021.

Kai Wang, Sanket Shah, Haipeng Chen, Andrew Perrault, Finale Doshi-Velez, and Milind Tambe. Learning mdps from features: Predict-then-optimize for sequential decision making by reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8795–8806, 2021.

Xiyao Wang, Wichayaporn Wongkamjan, and Furong Huang. Live in the moment: Learning dynamics model adapted to evolving policy. *arXiv preprint arXiv:2207.12141*, 2022.

Stefan Webb, Adam Golinski, Rob Zinkov, Tom Rainforth, Yee Whye Teh, Frank Wood, et al. Faithful inversion of generative models for effective amortized inference. *Advances in Neural Information Processing Systems*, 31, 2018.

Ran Wei, Siliang Zeng, Chenliang Li, Alfredo Garcia, Anthony McDonald, and Mingyi Hong. A bayesian approach to robust inverse reinforcement learning. In *Conference on Robot Learning*. PMLR, 2023.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33:15737–15749, 2020.

Shentao Yang, Shujian Zhang, Yihao Feng, and Mingyuan Zhou. A unified framework for alternating offline model training and policy learning. *arXiv preprint arXiv:2210.05922*, 2022.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020a.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020b.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Amy Zhang, Harsh Satija, and Joelle Pineau. Decoupling dynamics and reward for transfer learning. *arXiv preprint arXiv:1804.10689*, 2018.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.