Compositional Learning-based Planning for Vision POMDPs

Sampada Deglurkar[*,1]   SAMPADA_DEGLURKAR@BERKELEY.EDU
Michael H. Lim[*,1]   MICHAELHLIM@BERKELEY.EDU
Johnathan Tucker[2]   JOTU9759@COLORADO.EDU
Zachary N. Sunberg[2]   ZACHARY.SUNBERG@COLORADO.EDU
Aleksandra Faust[3]   FAUST@GOOGLE.COM
Claire J. Tomlin[1]   TOMLIN@EECS.BERKELEY.EDU
[1]*UC Berkeley,* [2]*CU Boulder,* [3]*Google Research.*

Abstract

The Partially Observable Markov Decision Process (POMDP) is a powerful framework for capturing decision-making problems that involve state and transition uncertainty. However, most current POMDP planners cannot effectively handle high-dimensional image observations prevalent in real world applications, and often require lengthy online training that requires interaction with the environment. In this work, we propose Visual Tree Search (VTS), a compositional learning and planning procedure that combines generative models learned offline with online model-based POMDP planning. The deep generative observation models evaluate the likelihood of and predict future image observations in a Monte Carlo tree search planner. We show that VTS is robust to different types of image noises that were not present during training and can adapt to different reward structures without the need to re-train. This new approach significantly and stably outperforms several baseline state-of-the-art vision POMDP algorithms while using a fraction of the training time.

Keywords: Partially Observable Markov Decision Process, Monte Carlo Tree Search, Compositional Learning, Generative Models

1. Introduction

Many sequential decision making problems, such as autonomous driving (Bai et al., 2015; Sunberg et al., 2017), cancer screening (Ayer et al., 2012), spoken dialog systems (Young et al., 2013), and aircraft collision avoidance (Holland et al., 2013), involve uncertainty in both sensing and planning. Planning under partial observability is challenging, as the agent must address both localization and uncertainty-aware planning through active information gathering in the environment. By capturing the observation uncertainty through a *belief* distribution over possible states, the agent will be able to fully close the observation-plan-action loop. While there are many methods that can either handle visual localization (Jonschkowski et al., 2018; Karkus et al., 2020, 2021) or planning under uncertainty (Van Den Berg et al., 2012; Todorov and Li, 2005), a naïve combination of these methods, for instance by assuming the certainty equivalence principle or simplifying the observation space, may not yield meaningful closed-loop control policies that enable active information gathering under more general environmental assumptions.

The partially observable Markov decision process (POMDP) formalism is a powerful framework that can capture and systematically solve these sequential decision making under uncertainty problems. However, finding an optimal POMDP policy is computationally demanding, and often
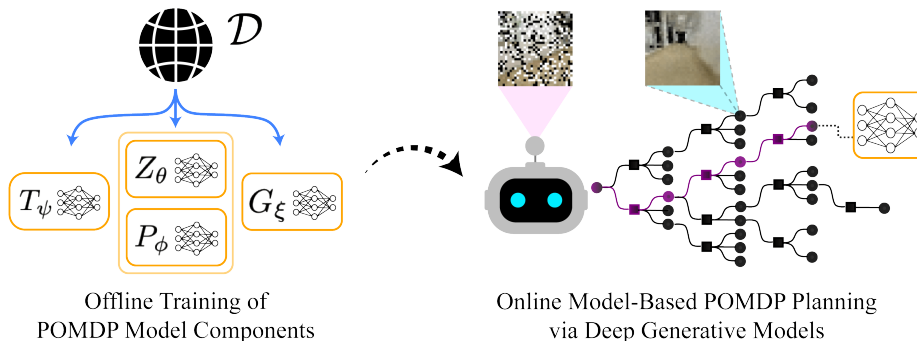
Figure 1: Visual Tree Search interfaces offline deep generative model training with online model-based POMDP filtering and planning.

intractable, due to the uncertainty introduced by imperfect observations (Papadimitriou and Tsitsiklis, 1987). One popular approach to deal with this challenge is to use *online* algorithms that look for local approximate policies as the agent interacts with the environment rather than a global policy that maps every possible outcome to an action, such as Monte Carlo tree search (MCTS) and similar variants (Browne et al., 2012; Silver and Veness, 2010; Sunberg and Kochenderfer, 2018; Ye et al., 2017; Kurniawati and Yadav, 2016). Many of these state-of-the-art MCTS algorithms enjoy computational efficiency (Sunberg and Kochenderfer, 2018; Mern et al., 2021; Lim et al., 2021) and finite sample convergence guarantees to the optimal policy (Lim et al., 2020, 2021). Despite their flexibility and optimality, these methods rely on having access to generative models and observation density models, which limits the class of problems they can solve in practice. In many realistic scenarios with high dimensional observations like RGB images, these POMDP methods cannot be applied without knowing or learning the relevant models or simplifying the environment.

Recently, there has been an increased interest in solving *vision POMDPs*, i.e. POMDPS with image or video observations, using deep learning methods. Model-free vision POMDP algorithms train an end-to-end deep neural network policy to learn both a latent belief representation and a planner (Karkus et al., 2017; Mnih et al., 2013; Igl et al., 2018), which benefit from not having to specify the transition and observation models and can learn complex policies. However, they may lack interpretability, not generalize well to new unseen tasks, and not leverage much prior knowledge about the system, especially in robotics settings. In contrast, model-based vision POMDP algorithms (Wang et al., 2020; Singh et al., 2021) combine classical filtering and planning techniques with deep learning. While such algorithmic structure allows the models to focus on specific tasks, making them sample efficient and robust, these methods often rely on simplified approaches for planning in the belief space and require learning an advantage function that only partially captures uncertainty by coupling observations with rewards.

Thus, we propose Visual Tree Search (VTS), a procedure to solve vision POMDPs by combining deep generative models and online tree search planning, effectively framing the POMDP reinforcement learning problem as a compositional unsupervised learning problem. Our key insight is to utilize compositional learning approaches to bridge the offline training of individual sets of models with online model-based planning that uses learning-enabled components. Additionally, we decouple observation and reward information during the online planning by sampling from the observation distribution. Introducing the algorithmic prior knowledge of particle filtering and MCTS decreases the computational complexity required by the learning and captures state and transition

uncertainty without dependence on reward structure. Our empirical analyses demonstrate that VTS enhances performance, robustness, and interpretability of neural network components.

2. Background

POMDPs. A POMDP is defined with a 7-tuple $(S, A, O, T, Z, R, \gamma)$, with state space S, action space A, observation space O, transition density $T(s'|s, a)$, observation density $Z(o|s)$, reward function $R(s, a)$, and discount $\gamma \in [0, 1)$ (Kochenderfer, 2015; Bertsekas, 2005). Specifically, we say "continuous POMDPs" to denote those with continuous state, action, and observation spaces. Since the agent receives only noisy observations of the true state, it can infer the state by maintaining a belief $b_t \in B$ at each step t and updating it with the new action and observation pair (a_{t+1}, o_{t+1}) via Bayesian filtering (Kaelbling et al., 1998). A policy $\pi : B \to A$ maps a belief b to an action a. The agent seeks to find an optimal policy π^* that maximizes the expected cumulative reward.

Monte Carlo Tree Search. In Monte Carlo planning, it is not always necessary to evaluate the exact probability of all transitions and observations, and merely generating samples of next state s', reward r, and observation o is sufficient. In particular, many Monte Carlo Tree Search (MCTS) algorithms only require that we have generative models that can generate (s', r, o) samples with densities $s' \sim T(s'|s, a)$ and $o \sim G(o|s')$, and observation density models that can evaluate the likelihood $w = Z(o|s')$. Using these samples, MCTS can reason about the transition and observation densities by balancing exploration and exploitation to approximate the true reward distribution. This results in an approximate online policy that maximizes the expected sum of rewards at each planning step.

Deep generative models. Deep generative models can be used to sample from probability distributions over high dimensional spaces such as spaces of images and videos. While some deep generative models, such as Generative Adversarial Networks (GANs), try to sample from distributions by making the samples seem as "realistic" as possible, other models such as Variational Autoencoders (VAEs) map complex distributions to simpler ones via latent space embedding. Conditional generative models condition on another variable to sample from conditional distributions (Mirza and Osindero, 2014; Zhao et al., 2017). In our MCTS planner, we use deep generative models G to sample observations conditioned on the state, and P to propose state particles given the current observation. For our neural network training procedures, we make the ground truth state information available to the planner during training, but keep it unavailable during testing.

3. Related Works

Planning under uncertainty. Planning under state and transition uncertainty requires planners to simultaneously localize and optimally plan through active information gathering. Popular control-based methods involve variants of the iterative Linear Quadratic Gaussian (Todorov and Li, 2005; Van Den Berg et al., 2012; Lee et al., 2017), which perform trajectory optimization over a simplified belief space. While such methods can handle continuous dynamics and leverage fast optimization computational tools, they cannot effectively handle image observations and high-dimensional state spaces. They also make simplifying Gaussian belief assumptions and are only locally optimal within the simplified belief space. Other sampling-based methods such as Sequential Monte Carlo (SMC) (Piché et al., 2019; Wang et al., 2020) ameliorate the above shortcomings by augmenting sampling-based planning with advantage networks, but do so at the cost of being unable to reason

about future observations. Such an approximation effectively assumes that the state uncertainty vanishes at the next step, which is proven to be sometimes suboptimal (Kaelbling et al., 1998).

On the other hand, state-of-the-art tree search planners have shown success in relatively large or continuous space POMDP planning problems. Most notably, POMCPOW and PFT-DPW (Sunberg and Kochenderfer, 2018), LABECOP (Hoerger and Kurniawati, 2020), and DESPOT-α (Garg et al., 2019) were shown to be effective in solving continuous observation POMDP problems. They use weighted collections of particles to efficiently represent complex beliefs. Provided the particles are weighted appropriately based on the observation likelihood, tree search using these particle beliefs will converge to a globally optimal policy (Lim et al., 2020). These tree search methods require access to generative models and observation density models to effectively plan, which we aim to learn with neural networks to extend the scope of the tree search methods. In this work, we integrate the PFT-DPW algorithm to handle learning-based model components.

Deep learning for vision POMDP. Recently, there has been increased interest in solving POMDPs involving visual observations through deep learning (Guillén et al., 2005; Karkus et al., 2018; Zhang et al., 2019). Model-free vision POMDP solvers such as QMDP-net (Karkus et al., 2017) and Deep Variational Reinforcement Learning (Igl et al., 2018) maintain a latent belief vector whose update rule is learned via a neural network. They then learn corresponding value or policy networks in this latent belief state and action space. Furthermore, since these methods usually make minimal sets of assumptions, extensions of such techniques can also be seen in embodied artificial intelligence applications (Karkus et al., 2021; Ai et al., 2022)

Model-based vision POMDP works such as Differentiable Particle Filter (DPF) (Jonschkowski et al., 2018) allow conventional particle filtering techniques to be interfaced with complex visual observations. This algorithm contains multiple linked neural network components, including a particle proposer, an observation model, and a dynamics model, that are designed to be trained end-to-end. A recent work extends DPF with entropy regularization and provides convergence guarantees (Corenflos et al., 2021). Dual Sequential Monte Carlo (DualSMC) (Wang et al., 2020) extends the DPF methods further to introduce an adversarial filtering objective and integrate in the SMC planner, making it a fully closed-loop POMDP solver. In this work, we extend DPF and DualSMC to interface tree search planners.

Compositional learning. In compositional learning, a learning task is broken down into neural network components that each specialize in different tasks. These components are then integrated to learn complex relations, allowing for less data and training resources to be used overall. Compositional learning can be achieved either through provided compositional structure in the form of an algorithmic prior (Andreas et al., 2016; Hudson and Manning, 2018), or by automatically discovering such structures (Rosenbaum et al., 2018; Alet et al., 2018; Kirsch et al., 2018; Meyerson and Miikkulainen, 2018). This paradigm has enjoyed success in various reinforcement learning settings, particularly multi-task problems (Devin et al., 2017; Ha and Schmidhuber, 2018; Yang et al., 2020; Mittal et al., 2020; Lim et al., 2022).

4. Visual Tree Search

In the Visual Tree Search (VTS) algorithm, we integrate the learned POMDP model components with classical filtering and planning techniques, interfaced by the belief state. This results in a POMDP solver that can learn model components that are more sample efficient and interpretable
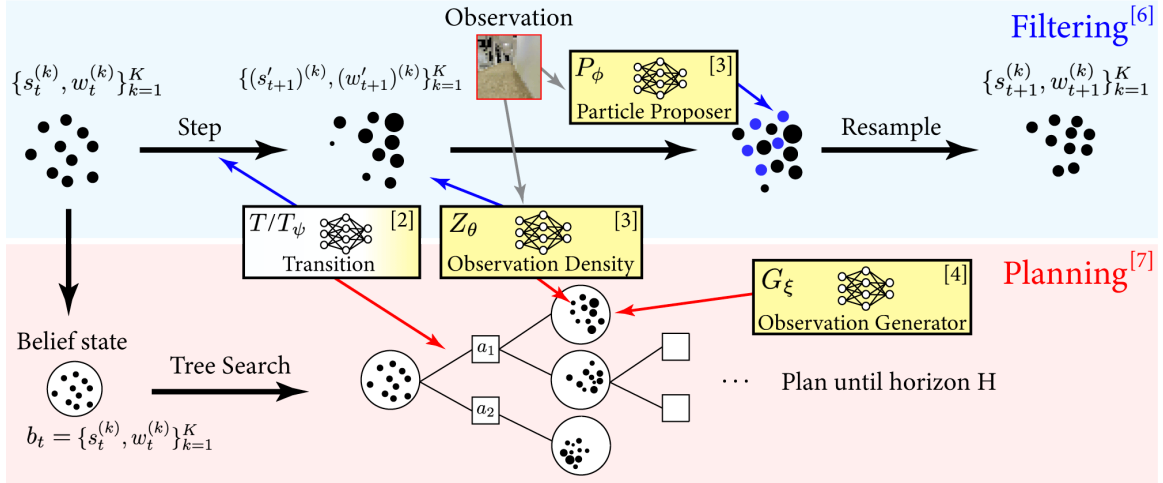
Figure 2: Overview of Visual Tree Search (VTS). The models learned with neural networks are shown in yellow – the transition model is shown in gradient, since it can be learned or pre-defined. The contributions of the models for filtering are shown in blue arrows, and planning in red arrows, and numbers in brackets show corresponding steps in Algorithm 1.

than end-to-end approaches. It also benefits from having a robust particle filter and planner built upon techniques with theoretical guarantees, and can adapt to different task rewards. To integrate planning techniques that use online tree search, we must have access to a conditional generative model that can generate image observations o from a given state s according to the likelihood density $Z(o|s)$. In this section, we outline the filtering and planning algorithms and models, and the compositional training procedure.

4.1. Differentiable Particle Filtering

For particle filtering, we leverage a family of architectures called Differentiable Particle Filters (DPF) (Jonschkowski et al., 2018), which combine classical particle filtering algorithms with convolutional neural networks that can handle complex observations. We learn two neural network-based models: (1) Observation density Z_θ that gives the likelihood weights $w_t = Z_\theta(o_t|s_t)$, (2) Particle proposer P_ϕ that allows us to sample states $s_t \sim P_\phi(o_t)$. The Greek letters denote the parameters of these neural network models. Specifically for our work, we adapt the DPF architecture introduced in DualSMC (Wang et al., 2020), in which the observation and proposer networks are trained with an adversarial optimization objective: Z_θ serves as a discriminator that gives higher likelihoods to states that are more likely for a given observation, and P_ϕ serves as a conditional generator that proposes plausible state particles for a given observation. In principle, we can also train DPF with entropy regularization (Corenflos et al., 2021) with optimality guarantees. We assume that the transition model T is known, which is not a limiting assumption for many POMDP problems (e.g. POMDPs with physical dynamics). However, in principle it can be learned in a supervised fashion from a dataset of (s_t, a_t, s_{t+1}) triplets using a simple regression model.

The filtering procedure is described below; the belief is represented by $b_t \approx \{s_t^{(k)}, w_t^{(k)}\}_{k=1}^K$, with K as the number of particles. First, the agent takes a step with an action chosen by the planner. Then, the agent updates the predicted states $(s'_{t+1})^{(k)}$ with the transition model T. The observation

---

Algorithm 1 Visual Tree Search Algorithm.

Input: Hyperparameters for neural networks (ζ), DPF (χ), MCTS (ρ), maximum time step $T_{\max}$.
Output: Action a_t at each step t.

1: Collect data $\mathscr{D}$ of tuples (s_t, a_t, o_t, s_{t+1}) through random sampling or exploration.
2: [Optional] Train the transition model T_ψ with data tuples (s_t, a_t, s_{t+1}).
3: [Training] Jointly train the observation density model Z_θ and particle proposer model P_ϕ with data tuples $(s_t, o_t, \{\hat{s}_{t,i}\})$, where $\{\hat{s}_{t,i}\}$ are particle estimates of s_t.
4: [Training] Train the observation conditional generator model G_ξ with data tuples (s_t, o_t).
5: **for** $t = 1$ to $T_{\max}$ **do**
6:     [Filtering] If $t = 1$, initialize the belief state b_t. Otherwise, after receiving o_t, update the belief state b_t with DPF($T_\psi, Z_\theta, P_\phi, \chi$).
7:     [Planning] Run MCTS($T_\psi, Z_\theta, G_\xi, \rho$) on the belief state b_t to obtain and perform action a_t.
8: **end for**

---

o_t obtained from the environment is fed into the observation density Z_θ, which provides the likelihood of an observation given the state. This is used to update the likelihood weights $(w'_{t+1})^{(k)}$ of each particle. In order to ensure robustness in the particle representation of the belief state, the particle proposer deep generative model P_ϕ proposes plausible state particles for a given observation, replacing some fraction of the particles. This fraction is made to decay exponentially over time.

4.2. Tree Search Planner

Monte Carlo Tree Search. For the online planner, we use the Particle Filter Trees-Double Progressive Widening (PFT-DPW) algorithm (Sunberg and Kochenderfer, 2018). PFT-DPW is a particle belief-based MCTS planner that is relatively easy to implement and efficiently vectorizes particle filtering. Additionally, a simplified version of PFT-DPW has optimality guarantees (Lim et al., 2020). However, any continuous POMDP tree search planner can be used instead.

Although PFT-DPW does not require a discrete action space, we discretize the action space for our navigational problems to simplify the computation without loss of generality. We allow the robotic agent to move in the 8 cardinal and diagonal directions with full thrust so that actions in the tree search are sufficiently distinct. While this means we work with a limited action space, we can ensure that we travel with full thrust to get to the goal faster and reduce the complexity of both planning and generating observations. However, we could also work with a continuous action space in principle. Furthermore, we provide PFT-DPW with a naïve rollout policy of actuating straight towards the goal and calculating the expected reward, which PFT-DPW can use as a reference and vastly improve upon.

Observation conditional generative model. Deep conditional generative models enable online model-based POMDP planning with images. To plan with a tree search planner, we need to be able to generate the next step states and observations, and evaluate the likelihood of the observations. With models from DPF, we can generate the next step state with transition model T and calculate the likelihood weight with observation density Z_θ, which are also used in the filtering procedure. Thus, we only additionally need to learn a deep generative model G_ξ that generates an observation o_t given a state s_t: $o_t \sim G_\xi(s_t)$. We use a Conditional Variational Autoencoder (CVAE) (Sohn et al., 2015) for this, where the state s_t is the conditional variable, since it had the most consistent training and performance in our experiments.

4.3. Compositional Training of Visual Tree Search

We train each neural network model with pre-collected data $\mathscr{D}$ containing tuples of (s_t, a_t, o_t, s_{t+1}), as shown in Lines 1-4 of Algorithm 1. The Z_θ and P_ϕ models are trained on random batches containing $(s_t, o_t, \{\hat{s}_{t,i}\})$, which are states, observations, and synthetic belief particle sets generated by sampling states from a normal distribution centered at s_t. The G_ξ model is similarly trained on state and observation pairs (s_t, o_t).

In this way, the VTS training procedure shifts the POMDP problem from a reinforcement learning problem to a compositional learning problem, in which each model in the POMDP is trained in an unsupervised fashion. This drastically decreases the problem complexity, as we have explicit control over the learning objective of each model and the schedule of the training. It also allows us to better approximate the data distribution via sampling or exploration, as opposed to an evolving on-policy planner distribution that often starts off poorly and provides heavily biased data.

5. Experiments

We compare VTS to other state-of-the-art vision POMDP algorithms, DualSMC (Wang et al., 2020), DVRL (Igl et al., 2018) and PlaNet (Hafner et al., 2019), on two benchmark vision POMDP problems. First, we tested our algorithm on the 2D Floor Positioning problem (Wang et al., 2020) to demonstrate that using the VTS learning and planning procedure can significantly decrease the training time. Then, we prepared our own version of the 3D Light-Dark problem in the Stanford Large-Scale 3D Indoor Spaces dataset (Armeni et al., 2017) to set up a more challenging navigation task that requires the agent to plan with realistic indoor building RGB images. We also performed ablation tests with the 3D Light-Dark experiment in which we varied the reward structure using spurious traps and the observation space using random visual occlusions. In each section, we calculate the results of the planner performances for 1000 testing episodes for Floor Positioning and 500 for 3D Light-Dark. For online planning speed, VTS takes around 0.26 seconds to plan for Floor Positioning and 0.80 seconds for 3D Light-Dark on average, while other planners require less than 0.05 seconds for both problems. The experimental summary figures are given in Fig. 3. [1]

5.1. Floor Positioning Problem

In this problem, a robotic agent is randomly placed around the center of either the top or the bottom floor, and it must infer its position by relying on a radar-like observation in all four cardinal directions, which bounces off the nearest wall. The top and bottom floors are indistinguishable within the "corridor states" of the hallways, but the robotic agent can take advantage of the "wall states" by traveling closer to the top or bottom walls of each floor, where it can receive different observations due to the different wall placements in each floor. The agent must reach the goal and avoid the trap, where the goal is the left end of the hallway in the top floor and right end in the bottom floor, and the trap is at the opposite side of the goal in each floor.

Planner comparison. Overall, VTS is the most successful among the three planners with reasonable online planning time and steps taken, while requiring less than half the training time. VTS training is fast since it does not rely on planner performance and only needs to be supplied with relevant state and observation batches. However, other online training algorithms require the policy to

---

1. In addition, the full tabular results summary, VTS training data details, hyperparameters and computation details, and source code link are in the appendices provided at https://arxiv.org/abs/2112.09456.
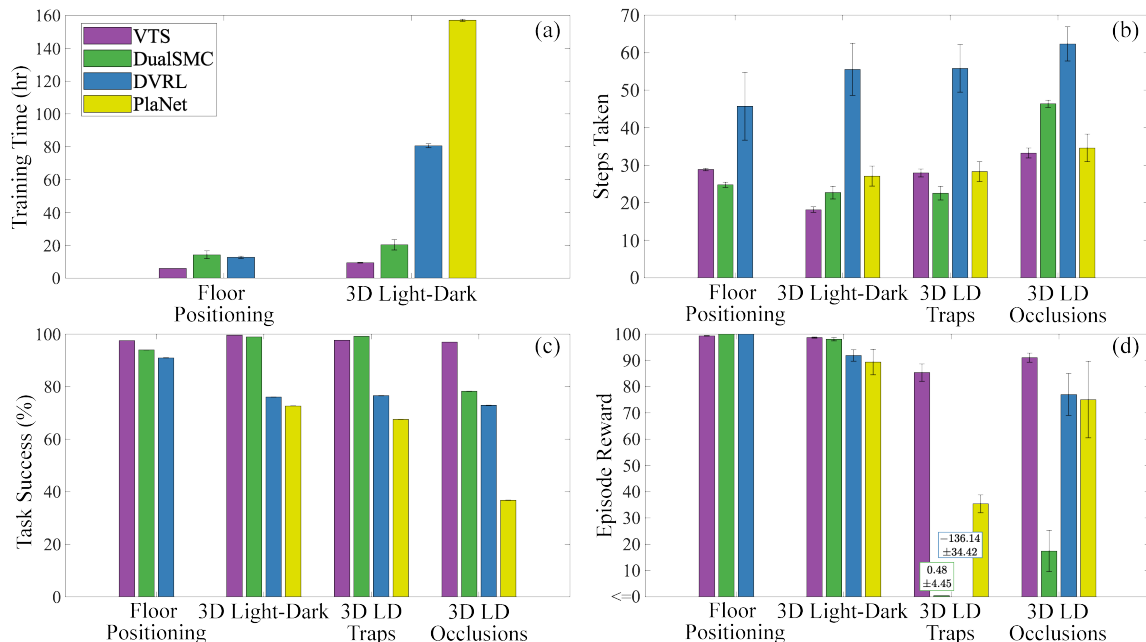
Figure 3: The results of 4 different planners, VTS, DualSMC, DVRL, and PlaNet, on Floor Positioning and 3D Light-Dark. Sub figures denote: (a) training time, (b) steps taken, (c) task success, and (d) episode reward.

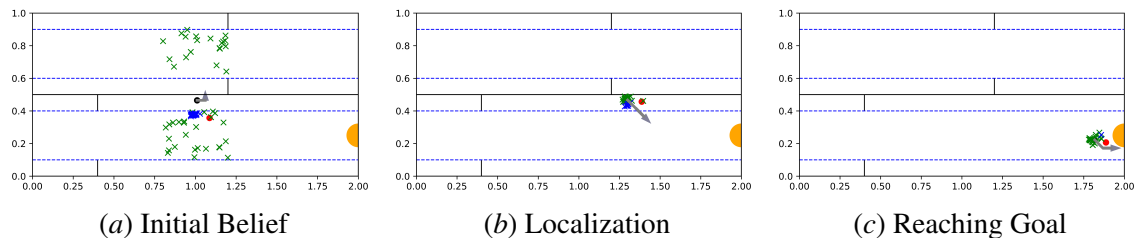(*a*) Initial Belief   (*b*) Localization   (*c*) Reaching Goal

Figure 4: Example VTS behavior for Floor Positioning, where the planner starts the problem with the initial belief (a), localizes in the wall states (b), and successfully reaches the goal (c).

interact with the environment, and as such, much of the time is spent earlier in the training episodes when the planner can neither localize nor reach the goal. Since tree search methods learn the policy online and are more difficult to parallelize, they typically trade off the offline training time with the online learning and planning time. Despite this, VTS plans reasonably fast while maintaining the efficiency and flexibility of an online planner. We also tested DualSMC with a known T model to ensure VTS is not given an advantage by knowing T for model-based planning, but observed no statistically significant difference in performance. In Fig. 4, we show an example trajectory of the VTS agent, in which the agent quickly localizes in the wall states and then reaches the goal.

5.2. 3D Light-Dark Problem

The Light-Dark problem is a family of problems in which an agent starting in a "dark" region can localize by taking a detour into a "light" region before reaching the goal. Observations are noisier

in the dark region than in the light region. The 3D Light-Dark problem extends this problem to have 3D image observations with salt-and-pepper pixel-wise noise.

The observations consist of $32 \times 32 \times 3$ RGB images from the agent's perspective. Our environment is more challenging to reason with than the one in Wang et al. (2020), since the RGB images are rendered from a realistic hallway scene dataset rather than a synthetic environment. The agent receives a reward for reaching the goal and a penalty if it enters a "trap".

Figure 5: Example schema for 3D Light-Dark, localizing in the light region to reach the goal in the dark region.

Planner comparison. In the Light-Dark problem and its variants, VTS performs the best among four planners not only by taking full advantage of the image features, but also by being able to adapt to different reward structures and distribution shifts in the noisy observations during test time. Also, DualSMC sometimes fails to have successful training seeds, and DVRL and PlaNet have lower task success rates despite all successful training seeds.

Among the successful training seeds for DualSMC, VTS and DualSMC have comparable suc-

Figure 6: VTS can localize without light observations (top), unlike DualSMC (bottom).

cess rates, but the VTS planner takes fewer steps to reach the goal. While this is only 4.5 steps difference on average, further inspection of the planner trajectories in Fig. 6 reveals an interesting insight. Unlike the traditional light-dark problems, the VTS results for 3D Light-Dark suggest that the planner in fact is able to localize solely with the noisy observations in the dark region. This shows that while the salt-and-pepper noise observations are hard to interpret, the corrupted RGB images actually contain sufficient information to localize given enough observations.

Test-time changes: Spurious traps. VTS has additional advantages in its robustness and adaptability, which we showcase through experiments on modified Light-Dark environments. First, to test the adaptability of different planners on test-time reward structure difference, we perform an ablation with spurious traps that appear during test time. These traps are regions of negative reward that do not affect the observation or transition distributions. They could, for example, represent the presence of unforeseen hazards. Over 500 testing episodes, we ran-

Figure 7: VTS can avoid new test time traps (top), while other planners like DVRL disregard the traps (bottom).

domly generate the locations of two 0.5×0.5 square traps over a particular strip in the environment and compare the performances of the models without additional training or modification.

Since VTS uses an online planner, it can easily adapt to new reward structures at test time without the need to retrain an entire planner. We see in Fig. 7 that while the other planners ignore these new trap regions because they are not represented by their models, the VTS planner is able to take into account rewards seen while planning. Thus, while the success rate of all planners remains
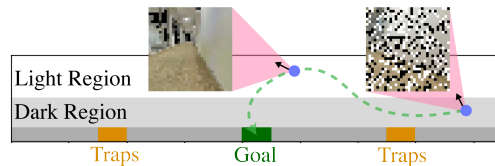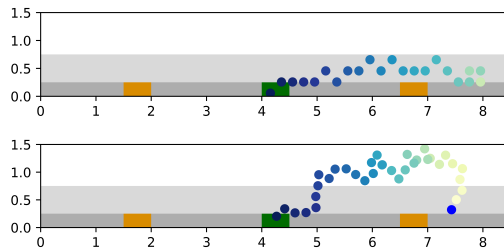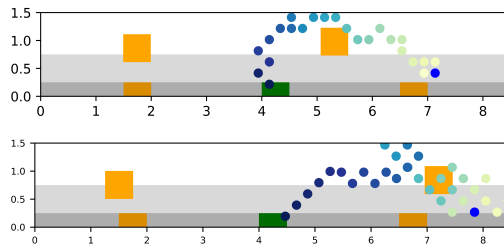
similar to the vanilla experiment, VTS achieves a much higher reward than all other planners while taking more steps as it actively tries to avoid these traps.

Test-time changes: Image occlusions. Second, we compare planner performances when the form of the noise in the image observations changes during test time. During test time only, images seen in the dark region contain random blacked out 15×15 squares instead of salt-and-pepper noise.

We also find that VTS is robust to distribution shifts in the image noise at test time. Fig. 8 shows VTS maintaining a good success rate and number of steps taken, while other planners are struggling to generalize to this new scenario. The additional robustness of VTS seems to be due to the Z_θ and P_ϕ models being trained with high fidelity data samples through random batch sampling or exploration.

In contrast, other planners that use on-policy training and complex non-separable architectures suffer from such distribution shifts. This is likely due to lack of control over on-policy exploration and
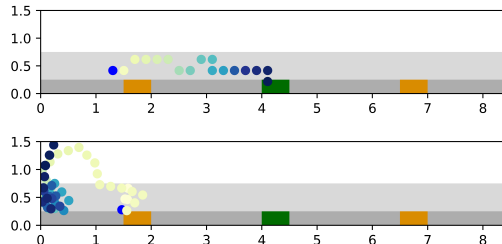

Figure 8: VTS can optimally plan even with occlusions (top), while other planners like PlaNet fail to reach the goal (bottom).

the inability to cover many possible scenarios with such limited interaction with the environment. Due to this difference in data variation, we conjecture that the Z_θ and P_ϕ models in VTS learn more precise and robust features.

5.3. Discussion

When comparing the performance of VTS with algorithms such as DVRL and PlaNet, we recognize the importance of explicitly modeling the connection between observations and states via belief particle sets. This comparison also demonstrates the ability of a compositional approach to allow for greater performance and robustness. We also demonstrate the difference made by using the G_ξ model to sample from the observation distribution $P(o|s)$ during planning by comparing VTS with DualSMC. Using G_ξ provides a way to decouple observation and reward information, unlike the advantage function that DualSMC uses during SMC-based planning. Additionally, VTS leverages offline training to reduce overall training time while still generating a strong policy.

6. Conclusion

The development of Visual Tree Search suggests new ways to think about integrating uncertainty-aware learning and control. VTS demonstrates that a more principled integration of control and planning techniques both illuminates the interpretability of each model component and saves training time by cleanly partitioning what each network is responsible for. This benefits researchers and practitioners alike, as a more interpretable and theoretically principled planner that is also quicker to train is beneficial in many practical safety critical scenarios with limited training resources.

Our work also raises the question of what is the most natural, effective, and safety-ensured method of combining pre-existing controllers and planners, which are extensively studied both theoretically and experimentally, with learning-based components, which can model many complex functions and phenomena. VTS is one such way of accomplishing that goal, but there are other ways to achieve principled integration of learning and control.

Acknowledgments

References

Bo Ai, Wei Gao, Vinay, and David Hsu. Deep visual navigation under partial observability. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9439–9446, 2022.

Ferran Alet, Tomas Lozano-Perez, and Leslie P. Kaelbling. Modular meta-learning. volume 87 of *PMLR*, pages 856–868. PMLR, 29–31 Oct 2018.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *IEEE CVPR*, June 2016.

I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, February 2017.

Turgay Ayer, Oguzhan Alagoz, and Natasha K. Stout. A POMDP approach to personalize mammography screening decisions. *Operations Research*, 60(5):1019–1034, 2012.

Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Sun Lee. Intention-aware online POMDP planning for autonomous driving in a crowd. In *IEEE ICRA*, pages 454–460, Seattle, WA, USA, 2015. IEEE.

D. Bertsekas. *Dynamic Programming and Optimal Control*. Massachusetts: Athena Scientific, 2005.

C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE T-CIAIG*, 4(1):1–43, 2012.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *ICML*, volume 139 of *PMLR*, pages 2100–2111. PMLR, 18–24 Jul 2021.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *IEEE ICRA*, pages 2169–2176. IEEE, 2017.

Neha P. Garg, David Hsu, and Wee Sun Lee. DESPOT-α: Online POMDP planning with large state and observation spaces. In *RSS*, 2019.

María Elena López Guillén, Luis Miguel Bergasa, Rafael Barea, and María Soledad Escudero. A navigation system for assistant robots using visually augmented pomdps. *Auton. Robots*, 19(1): 67–87, 2005.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *NeuRIPS*, pages 2451–2463. Curran Associates, Inc., 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565, 2019.

Marcus Hoerger and Hanna Kurniawati. An on-line POMDP solver for continuous observation spaces. *ArXiv e-prints*, 2020.

Jessica E. Holland, Mykel J. Kochenderfer, and Wesley A. Olson. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. *Air Traffic Control Quarterly*, 21(3):275–297, 2013.

Drew Arad Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *ICLR*, 2018.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. In *ICML*, volume 80 of *PMLR*, pages 2117–2126. PMLR, 10–15 Jul 2018.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *RSS*, Pittsburgh, Pennsylvania, June 2018.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99 – 134, 1998.

Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In *NeuRIPS*, volume 30. Curran Associates, Inc., 2017.

Peter Karkus, David Hsu, and Wee Sun Lee. Integrating algorithmic planning and deep learning for partially observable navigation. *ArXiv e-prints*, 2018.

Peter Karkus, Anelia Angelova, Vincent Vanhoucke, and Rico Jonschkowski. Differentiable mapping networks: Learning structured map representations for sparse visual localization. In *IEEE ICRA*, pages 4753–4759, 2020.

Peter Karkus, Shaojun Cai, and David Hsu. Differentiable slam-net: Learning particle slam for visual navigation. In *IEEE CVPR*, pages 2815–2825, June 2021.

Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. In *NeuRIPS*, volume 31. Curran Associates, Inc., 2018.

Mykel J. Kochenderfer. *Decision Making Under Uncertainty: Theory and Application*. Massachusetts: MIT Press, 2015.

Hanna Kurniawati and Vinay Yadav. An online POMDP solver for uncertainty planning in dynamic environment. In *Robotics Research*, pages 611–629. Springer, 2016.

Gilwoo Lee, Siddhartha S. Srinivasa, and Matthew T. Mason. GP-ILQG: Data-driven robust optimal control for uncertain nonlinear dynamical systems. *ArXiv e-prints*, 2017.

Michael H. Lim, Claire Tomlin, and Zachary N. Sunberg. Sparse tree search optimality guarantees in POMDPs with continuous observation spaces. In *IJCAI*, pages 4135–4142. International Joint Conferences on Artificial Intelligence, Inc., 7 2020.

Michael H. Lim, Claire J. Tomlin, and Zachary N. Sunberg. Voronoi progressive widening: Efficient online solvers for continuous state, action, and observation POMDPs. In *IEEE CDC*, pages 4493–4500, 2021.

Michael H. Lim, Andy Zeng, Brian Ichter, Maryam Bandari, Erwin Coumans, Claire Tomlin, Stefan Schaal, and Aleksandra Faust. Multi-task learning with sequence-conditioned transporter networks. In *IEEE ICRA*, pages 2489–2496, 2022.

John Mern, Anil Yildiz, Zachary Sunberg, Tapan Mukerji, and Mykel J. Kochenderfer. Bayesian optimized Monte Carlo planning. In *AAAI*. AAAI Press, 2021.

Elliot Meyerson and Risto Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *ICLR*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv e-prints*, 2014.

Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *ICML*, volume 119 of *PMLR*, pages 6972–6986. PMLR, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NeurIPS Deep Learning Workshop*. 2013.

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential monte carlo methods. In *ICLR*, 2019.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *ICLR*, 2018.

David Silver and Joel Veness. Monte-Carlo planning in large POMDPs. In *NeuRIPS*, pages 2164–2172. Curran Associates, Inc., 2010.

Gautam Singh, Skand Peri, Junghyun Kim, Hyunseok Kim, and Sungjin Ahn. Structured world belief for reinforcement learning in POMDP. In *ICML*, volume 139, pages 9744–9755. PMLR, 18–24 Jul 2021.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NeuRIPS*, volume 28. Curran Associates, Inc., 2015.

Zachary Sunberg and Mykel J. Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. In *ICAPS*, Delft, Netherlands, 2018. AAAI Press.

Zachary N. Sunberg, Christopher J. Ho, and Mykel J. Kochenderfer. The value of inferring the internal state of traffic participants for autonomous freeway driving. In *ACC*, pages 3004–3010, Seattle, WA, USA, 2017. IEEE.

E. Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *ACC*, volume 1, pages 300–306, 2005.

Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *International Journal of Robotics Research*, 31(11): 1263–1278, 2012.

Yunbo Wang, Bo Liu, Jiajun Wu, Yuke Zhu, Simon S. Du, Li Fei-Fei, and Joshua B. Tenenbaum. Dualsmc: Tunneling differentiable filtering and planning under continuous pomdps. In *IJCAI*, pages 4190–4198. International Joint Conferences on Artificial Intelligence Organization, 7 2020.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *ArXiv e-prints*, 2020.

Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. DESPOT: Online POMDP planning with regularization. *Journal of Artificial Intelligence Research*, 58:231–266, 2017.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. POMDP-based statistical spoken dialog systems: A review. *IEEE*, 101(5):1160–1179, 2013.

Marvin Zhang, Sharad Vikram, Laura M. Smith, Pieter Abbeel, Matthew J. Johnson, and Sergey Levine. SOLAR: deep structured representations for model-based reinforcement learning. In *ICML*, volume 97, pages 7444–7453, 2019.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *ArXiv e-prints*, 2017.