Mode-Constrained Model-Based Reinforcement Learning via Gaussian Processes: Supplementary Materials

A Dynamics Model

A.1 Sparse Gaussian Procesess

Each dynamics mode's predictions conditioned on its inducing variables follows from the properties of mutivariate normals and are given by,

$$p(\Delta \mathbf{s}_{t+1} \mid f_k(\boldsymbol{\zeta}_k)) = \mathbb{E}_{p(f_k(\hat{\mathbf{s}}_t) \mid f_k(\boldsymbol{\zeta}_k))} \left[p(\Delta \mathbf{s}_{t+1} \mid f_k(\hat{\mathbf{s}}_t)) \right] \tag{14}$$

$$p(f_k(\hat{\mathbf{s}}_t) \mid f_k(\boldsymbol{\zeta}_k)) = \mathcal{N}\left(f_k(\hat{\mathbf{s}}_t) \mid k_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) k_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} f_k(\boldsymbol{\zeta}_k), k_k(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_t) - k_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) k_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} k_k(\boldsymbol{\xi}, \hat{\mathbf{s}}_t) \right), \tag{15}$$

Similarly for the gating network we have,

$$\Pr(\alpha_t = k \mid \mathbf{h}(\boldsymbol{\xi})) = \mathbb{E}_{p(\mathbf{h}(\hat{\mathbf{s}}_t) \mid \mathbf{h}(\boldsymbol{\xi}))} \left[\Pr\left(\alpha_t = k \mid \mathbf{h}(\hat{\mathbf{s}}_t)\right) \right] \tag{16}$$

$$p(\mathbf{h}(\hat{\mathbf{s}}_t) \mid \mathbf{h}(\boldsymbol{\xi})) = \prod_{k=1}^{K} \mathcal{N}\left(h_k(\hat{\mathbf{s}}_t) \mid \hat{k}_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) \hat{k}_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} h_k(\boldsymbol{\xi}), \hat{k}_k(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_t) - \hat{k}_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) \hat{k}_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \hat{k}_k(\boldsymbol{\xi}, \hat{\mathbf{s}}_t) \right), \tag{17}$$

Predictive posteriors As each GP's inducing variables are normally distributed, the functional form of our predictive posteriors are given by,

$$p(f_k(\hat{\mathbf{s}}_t) \mid \hat{\mathbf{s}}_t, \mathcal{D}_{0:i}) \approx \int p(f_k(\hat{\mathbf{s}}_t) \mid f_k(\boldsymbol{\zeta}_k)) q(f_k(\boldsymbol{\zeta}_k)) \mathrm{d} f_k(\boldsymbol{\zeta}_k) = \mathcal{N}\left(f_k(\hat{\mathbf{s}}_t) \mid \mathbf{A}_k \mathbf{m}_k, k_k(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_t) + \mathbf{A}_k(\mathbf{S}_k - k_k(\boldsymbol{\xi}, \boldsymbol{\xi})) \mathbf{A}_k^T \right)$$

$$p(\mathbf{h}(\mathbf{s}_t) \mid \mathbf{s}_t, \mathcal{D}_{0:i}) \approx \prod_{k=1}^{K} q(h_k(\boldsymbol{\xi})) = \prod_{k=1}^{K} \mathcal{N}\left(h_k(\hat{\mathbf{s}}_t) \mid \hat{\mathbf{A}}_k \hat{\mathbf{m}}_k, \hat{k}_k(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_t) + \hat{\mathbf{A}}_k(\hat{\mathbf{S}}_k - \hat{k}_k(\boldsymbol{\xi}, \boldsymbol{\xi})) \hat{\mathbf{A}}_k^T \right),$$

where $\mathbf{A}_k = k_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) k_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1}$ and $\hat{\mathbf{A}}_k = \hat{k}_k(\hat{\mathbf{s}}_t, \boldsymbol{\xi}) \hat{k}_k(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1}$. Importantly, our predictive posteriors marginalise the inducing variables in closed form, with Gaussian convolutions.

Given our GP-based gating network, we are able to model the joint distribution over the gating function values $h_{k^*}(\bar{\mathbf{s}})$ along a trajectory $\bar{\mathbf{s}}$ with,

$$p(h_{k^*}(\bar{\mathbf{s}}) \mid \bar{\mathbf{s}}, \mathcal{D}_{0:i}) \approx q(h_{k^*}(\bar{\mathbf{s}})) = \mathcal{N}\left(h_{k^*}(\bar{\mathbf{s}}) \mid \boldsymbol{\mu}_{k^*}(\bar{\mathbf{s}}), \boldsymbol{\Sigma}^2_{k^*}(\bar{\mathbf{s}}, \bar{\mathbf{s}}) \right) \tag{18}$$

where $\boldsymbol{\mu}_{k^*}(\cdot)$ and $\boldsymbol{\Sigma}^2_{k^*}(\cdot, \cdot)$ are sparse GP mean and covariance functions, given by,

$$\boldsymbol{\mu}_{k^*}(\bar{\mathbf{s}}) = \hat{k}_{k^*}(\bar{\mathbf{s}}, \boldsymbol{\xi}) \hat{k}_{k^*}(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \hat{\mathbf{m}}_{k^*} \tag{19}$$

$$\boldsymbol{\Sigma}^2_{k^*}(\bar{\mathbf{s}}, \bar{\mathbf{s}}) = \hat{k}_{k^*}(\bar{\mathbf{s}}, \bar{\mathbf{s}}) + \hat{k}_{k^*}(\bar{\mathbf{s}}, \boldsymbol{\xi}) \hat{k}_{k^*}(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \left(\hat{\mathbf{S}}_{k^*} - \hat{k}_{k^*}(\boldsymbol{\xi}, \boldsymbol{\xi}) \right) \hat{k}_{k^*}(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \hat{k}_{k^*}(\boldsymbol{\xi}, \bar{\mathbf{s}}), \tag{20}$$

where $\hat{k}_{k^*}$ and $\boldsymbol{\xi}$ are the kernel and inducing inputs associated with the desired mode's gating function respectively. This sparse approximation arises because our dynamical model uses sparse GPs and approximates the posterior with,

$$q(h_{k^*}(\bar{\mathbf{s}})) = \int p(h_{k^*}(\bar{\mathbf{s}}) \mid h_{k^*}(\boldsymbol{\xi})) q(h_{k^*}(\boldsymbol{\xi})) \mathrm{d} h_{k^*}(\boldsymbol{\xi}), \tag{21}$$

where $q(h_{k^*}(\boldsymbol{\xi})) = \mathcal{N}\left(h_{k^*}(\boldsymbol{\xi} \mid \hat{\mathbf{m}}_{k^*}, \hat{\mathbf{S}}_{k^*} \right)$.

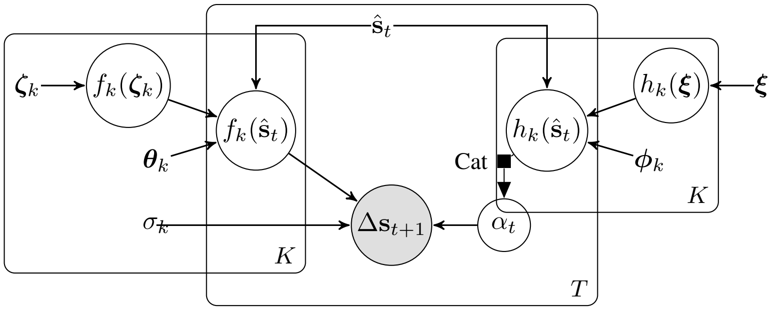

Figure 5: Graphical model of our augmented dynamics model where each state difference output $\Delta\mathbf{s}_{t+1}$ is generated by mapping the state-action input $\hat{\mathbf{s}}_t$ through the latent processes. The dynamics modes are shown on the left and the gating network on the right. The generative processes involve evaluating the gating network and sampling a mode indicator variable α_t. The indicated mode's latent function f_k and noise model $\mathcal{N}(0, \sigma_k^2)$ are then evaluated to generate the output $\Delta\mathbf{s}_{t+1}$.

A.2 Gating network

Bernoulli ($K = 2$) Instantiating the model with two experts, $\alpha_t \in \{1, 2\}$, is a special case where only a single gating function is needed. This is because the output of a function $h(\hat{\mathbf{s}}_t)$ can be mapped through a sigmoid function sig : $\mathbb{R} \rightarrow [0, 1]$ and interpreted as a probability,

$$\Pr(\alpha_t = 1 \mid h(\hat{\mathbf{s}}_t)) = \text{sig}(h(\hat{\mathbf{s}}_t)). \tag{22}$$

If this sigmoid function satisfies the point symmetry condition then the following holds, $\Pr(\alpha_t = 2 \mid h(\hat{\mathbf{s}}_t)) = 1 - \Pr(\alpha_t = 1 \mid h(\hat{\mathbf{s}}_t))$. This only requires a single gating function and no normalisation term needs to be calculated. If the sigmoid function in Eq. (22) is selected to be the Gaussian cumulative distribution function $\Phi(h(\cdot)) = \int_{-\infty}^{h(\cdot)} \mathcal{N}(\tau|0, 1)\mathrm{d}\tau$, then the mixing probability can be calculated in closed-form,

$$\begin{aligned}\Pr(\alpha_t = 1 \mid \hat{\mathbf{s}}_t) &= \int \Phi\left(h(\hat{\mathbf{s}}_t)\right) \mathcal{N}\left(h(\hat{\mathbf{s}}_t) \mid \mu_h, \sigma_h^2\right) \mathrm{d}h(\hat{\mathbf{s}}_t) \\ &= \Phi\left(\frac{\mu_h}{\sqrt{1 + \sigma_h^2}}\right),\end{aligned} \tag{23}$$

where μ_h and σ_h^2 represent the mean and variance of the gating GP at $\hat{\mathbf{s}}_t$ respectively.

Softmax ($K > 2$) In the general case, when there are more than two experts, the gating network's likelihood is defined as the Softmax function,

$$\Pr\left(\alpha_t = k \mid \mathbf{h}(\hat{\mathbf{s}}_t)\right) = \text{softmax}_k\left(\mathbf{h}(\hat{\mathbf{s}}_t)\right) = \frac{\exp\left(h_k(\hat{\mathbf{s}}_t)\right)}{\sum_{j=1}^{K} \exp\left(h_j(\hat{\mathbf{s}}_t)\right)}. \tag{24}$$

Each mode's mixing probability $\Pr(\alpha_t = k \mid \hat{\mathbf{s}}_t)$ is then obtained by marginalising **all** of the gating functions. In the general case where $\Pr\left(\alpha_t = k \mid \mathbf{h}(\hat{\mathbf{s}}_t)\right)$ uses the softmax function in Eq. (24), this integral is intractable, so we approximate it with Monte Carlo quadrature.

B Illustrative Example

ModeRL is tested on a 2D quadcopter navigation example shown in Fig. 6. The goal is to fly the quadcopter from an initial state $\mathbf{s}_0$, to a target state $\mathbf{s}_f$ (white star). However, it considers a quadcopter operating in an environment subject to spatially varying wind – induced by a fan – where two dynamics modes can represent the system,

Mode 1 is an *operable* dynamics mode away from the fan,

Mode 2 is an *inoperable*, turbulent dynamics mode in front of the fan.

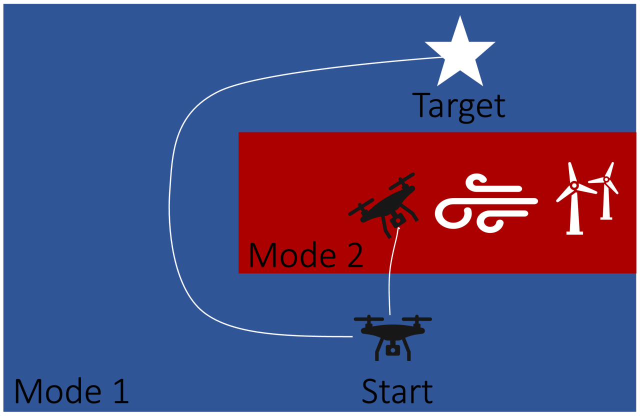

Figure 6: **Mode-constrained quadcopter navigation** - Diagram showing a top-down view of a quadcopter subject to two dynamics modes: 1) an *operable* dynamics mode (blue) and 2) an *inoperable*, turbulent dynamics mode induced by a strong wind field (red). The goal is to navigate to the target state $\mathbf{s}_f$ (white star), whilst avoiding the turbulent dynamics mode (red).

The turbulent dynamics mode is subject to higher drift (in the negative x direction) and to higher diffusion (transition noise). It is hard to know the exact turbulent dynamics due to complex and uncertain interactions between the quadcopter and the wind field. Further to this, controlling the system in the turbulent dynamics mode may be infeasible. This is because the unpredictability of the turbulence may cause catastrophic failure. Therefore, when flying the quadcopter to the target state $\mathbf{s}_f$, it is desirable to find trajectories that avoid entering this turbulent dynamics mode.

The state-space of the velocity controlled quadcopter example consists of the 2D Cartesian coordinates $\mathbf{s} = (x, y)$ and the controls consist of the speed in each direction, given by $\mathbf{a} = (v_x, v_y)$.

The reward function is given by,

$$r(\mathbf{s}_t, \mathbf{a}_t) = -\left(\mathbf{s}_t - \mathbf{s}_f\right)^T \mathbf{Q} \left(\mathbf{s}_t - \mathbf{s}_f\right) - \mathbf{a}_t^T \mathbf{R} \mathbf{a}_t \tag{25}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are user-defined, real symmetric positive semi definite and positive definite weight matrices respectively. In our experiments we set both $\mathbf{Q}$ and $\mathbf{R}$ to be the identify matrix.

C Experiment Configuration

This section details how the experiments were configured.

Initial data set $\mathcal{D}_0$ The initial data set was collected by simulating 50 random trajectories with horizon $T = 15$ from the start state $\mathbf{s}_0 = \{x_0, y_0\}$ and terminating them when they left the initial state domain $\mathcal{S}_0 = \{\mathbf{s} \in \mathcal{S} \mid x_0 - 1 < x < x_0 + 1, y_0 - 1 < y < y_0 + 1\}$.

Model learning In all experiments the dynamics model was instantiated with $K = 2$ modes. Each mode's dynamics GP used a separate Matern 5/2 kernel with ARD for each output dimension d but shared it's inducing variables for each output dimension d. Further to this, each mode learned separate constant mean function values and separate noise variances for each output dimension. The gating network used a single gating function with a Matern 5/2 kernel with ARD and a zero mean function. An early stopping callback was used to terminate the dynamics model's training. The early stopping callback used a min delta of 0 and a patience of 50. This meant that training terminated after 50 epochs of no improvement. All of the dynamics model's initial parameters are shown in Table 1.

Policy In all experiments, ModeRL used a horizon of $T = 15$ and was configured with $\delta = 0.2$. At each episode, ModeRL uses the previous solution as the initial solution for the trajectory optimiser.

Table 1: Experiment configuratin and parameter settings.

	Description	Symbol	Value
Environment	Start state	$\mathbf{s}_0$	$[2.0, -2.5]$
	Target state	$\mathbf{s}_f$	$[1.2, 3.0]$
	State reward weight	$\mathbf{Q}$	$\mathrm{diag}([1, 1])$
	Control reward weight	$\mathbf{R}$	$\mathrm{diag}([1, 1])$
Policy π	Mode constraint	δ	0.2
	Horizon	T	15
	Exploration weight	β	10.0
Dynamics optimiser	Batch size	N_b	64
	Num epochs	N/A	20000
	Num gating samples	N/A	1
	Num expert samples	N/A	1
	Learning rate	N/A	0.01
	Epsilon	N/A	1×10^{-8}
Dynamics mode 1 f_1	Constant mean function	c_{f_1}	$[0, 0]$
	Kernel variance ($d = 1$)	$\sigma^2_{f_1}$	1
	Kernel lengthscales ($d = 1$)	l_{f_1}	$[1, 1, 1, 1]$
	Kernel variance ($d = 2$)	$\sigma^2_{f_1}$	1
	Kernel lengthscales ($d = 2$)	l_{f_1}	$[1, 1, 1, 1]$
	Likelihood variance	σ^2_1	$\mathrm{diag}([1, 1])$
	Num inducing points	M	50
	Inducing inputs	$\boldsymbol{\zeta}_1$	$\boldsymbol{\zeta}_1 \subseteq \hat{\mathbf{S}}_0$ with $\#\boldsymbol{\zeta}_1 = M$
Dynamics mode 2 f_2	Constant mean function	c_{f_2}	$[0, 0]$
	Kernel variance ($d = 1$)	$\sigma^2_{f_2}$	1
	Kernel lengthscales ($d = 1$)	l_{f_2}	$[1, 1, 1, 1]$
	Kernel variance ($d = 2$)	$\sigma^2_{f_2}$	1
	Kernel lengthscales ($d = 2$)	l_{f_2}	$[1, 1, 1, 1]$
	Likelihood variance	σ^2_2	$\mathrm{diag}([1, 1])$
	Num inducing points	M	50
	Inducing inputs	$\boldsymbol{\zeta}_2$	$\boldsymbol{\zeta}_2 \subseteq \hat{\mathbf{S}}_0$ with $\#\boldsymbol{\zeta}_2 = M$
Gating function 1 h_1	Kernel variance	$\sigma^2_{h_1}$	1
	Kernel lengthscales	l_{h_1}	$[0.8, 0.8]$
	Active dims	N/A	$[0, 1]$
	Num inducing points	M_h	90
	Inducing inputs	$\boldsymbol{\xi}$	$\boldsymbol{\xi} \subseteq \hat{\mathbf{S}}_0$ with $\#\boldsymbol{\xi} = M_h$