# On Uncertainty in Deep State Space Models for Model-Based Reinforcement Learning

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Improved state space models, such as *Recurrent State Space Models (RSSMs)*, are a key factor behind recent advances in model-based reinforcement learning (RL). Yet, despite their empirical success, many of the underlying design choices are not well understood. We show that *RSSMs* use a sub-optimal inference scheme leading to an over-estimation of aleatoric uncertainty. We analyze this inference scheme and find that, while being sub-optimal for prediction, it implicitly regularizes *RSSMs*, allowing them to succeed in model-based RL. We postulate that this implicit regularization fulfills the same functionality as explicitly modeling epistemic uncertainty, which is crucial for many other model-based RL approaches. Yet, the sub-optimal inference also makes analyzing or improving *RSSMs* difficult as the beneficial over-estimated aleatoric uncertainty is a side-effect of the inference scheme and not a result of optimizing first order principles. Thus, we propose an alternative approach building on well-understood components for modeling aleatoric and epistemic uncertainty, dubbed *Variational Recurrent Kalman Network (VRKN)*. This approach uses Kalman updates for exact smoothing inference in a latent space and Monte Carlo Dropout to model epistemic uncertainty. Due to the Kalman updates, the *VRKN* naturally handles varying numbers of observations per time step. *VRKN*-based agents match the performance of *RSSM*-based agents in the deterministic standard benchmarks while outperforming them in tasks where appropriately capturing uncertainty in states and observations is crucial.

## 1 Introduction

Accurate system models are crucial for model-based control and reinforcement learning (RL) in autonomous systems applications under partial observability. Practitioners commonly use state space models (SSMs) (Murphy, 2012) to formalize such systems. SSMs consist of a dynamics model, describing how one state relates to the next, and an observation model, which describes how system states generate observations. Yet, dynamics and observation models are unknown for most relevant problems, and exact inference in the resulting SSM is usually intractable. Researchers have proposed numerous approaches to learn the models from data and approximate the inference to solve those issues.

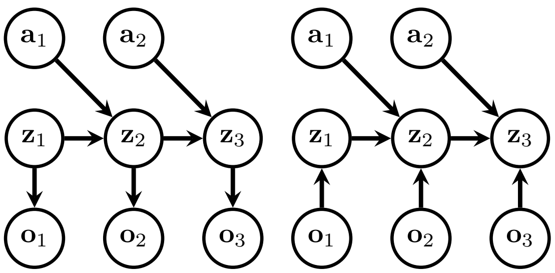
*Recurrent State Space Models (RSSMs)* (Hafner et al., 2019) are of particular interest here. Using *RSSMs* as the backbone for their *Deep Planning Network (PlaNet)*, Hafner et al. (2019) showed that variational latent dynamics learning can succeed in image-based RL for complex control tasks. Combined with simple planning, *RSSMs* can match the performance of model-free RL while requiring significantly fewer environment interactions. The authors later improved upon their original model, including a parametric policy trained on imagined trajectories (*Dreamer*) (Hafner et al., 2020). In general, approaches based on *RSSM*s have found considerable interest in the model-based RL community. Yet, while *RSSM*s clearly draw inspiration from classical SSMs, they use a simplified inference scheme. During inference, they assume the belief is independent of future observations instead of using the correct smoothing assumptions (Murphy, 2012) to obtain the belief. We formalize this observation in Section 2 and discuss how these simplified assumptions result in a theoretically looser variational lower bound. Further, we analyze the assumptions' effects on model learning, where they cause an overestimation of aleatoric uncertainty, i.e., uncertainty due to the inherent stochasticity of the system.

The *RSSM's* inference assumptions are a double-edged sword for model-based RL. On the one hand, the overestimated aleatoric uncertainty can be beneficial for model-based RL as it leads to dynamics models that generalize better and are more robust to *objective mismatch* (Luo et al., 2019; Lambert et al., 2020). Such *objective mismatch* arises because model-based RL approaches use a different loss and data distribution for training than for evaluation and data collection. While many approaches rely on explicit epistemic uncertainty to tackle this issue (Chua et al., 2018; Janner et al., 2019), *RSSMs* succeed without capturing epistemic uncertainty. On the other hand, they complicate the design and analysis of the system as the overestimated aleatoric uncertainty is a side-product of the inference scheme and does not follow first-order principles. As already reported by Hafner et al. (2019) purely stochastic models do not yield satisfactory results. Thus, the *RSSM* relies on a *deterministic-path*, combining stochastic and deterministic features to form the latent state. Further, as we will show in our experiments, overestimating the aleatoric uncertainty can lead to poor performance in settings where correctly estimating it is important.

We show that removing this issue for *RSSMs* by implementing a naive approach to smoothing yields unsatisfactory results, even if the model uses explicit epistemic uncertainty to compensate for the missing over-estimation of the aleatoric uncertainty. To make smoothing inference work, we redesign the model architecture from first principles and propose the *Variational Recurrent Kalman Network (VRKN)* which combines well-understood components for aleatoric and epistemic uncertainty. It uses a latent linear-Gaussian parameterization, allowing closed-form smoothing inference in the latent space and proper estimation of aleatoric uncertainty. Further, it does not require an additional deterministic path, yielding a purely stochastic model. Finally, we introduce a Bayesian treatment of our transition model's parameters, explicitly modeling the system's epistemic uncertainty, using Monte Carlo Dropout (Gal & Ghahramani, 2016). The resulting architecture allows model-based agents to perform comparably to *RSSM*-based agents in deterministic environments. If the tasks require accurate estimation of aleatoric uncertainty, the *VRKN* improves the agents' performance. Due to its linear Gaussian formulation, the *VRKN* can naturally deal with missing observations and fuse information from multiple sensors working at different frequencies, which is useful in many realistic applications. Here, observations such as camera images are only available at low frequencies, while other inputs, e.g., proprioceptive measurements, are available at a much higher frequency. Finally, our approach can serve as a basis for further improvements in this direction as we can now independently consider, adapt, and improve the components for aleatoric and epistemic uncertainty.

## 2 Inference and Learning in State Space Models

State Space Models (SSMs)(Murphy, 2012) assume a sequence of observations $\mathbf{o}_{\leq T} = \{\mathbf{o}_t\}_{t=0\cdots T}$ is generated by a sequence of latent variables $\mathbf{z}_{\leq T} = \{\mathbf{z}_t\}_{t=0\cdots T}$, given a sequence of actions $\mathbf{a}_{\leq T} = \{\mathbf{a}_t\}_{t=0\cdots T}$. In SSMs, each observation $\mathbf{o}_t$ is assumed to only depend on the current latent state $\mathbf{z}_t$ via an observation model $p(\mathbf{o}_t|\mathbf{z_t})$. Further, they assume the latent states are Markovian, i.e., each latent state only depends on its direct predecessor and the corresponding action via a dynamics model $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{a}_t)$. Finally, the initial state is distributed according to a distribution $p(\mathbf{z}_0)$. Figure 1a shows the corresponding graphical models. Typically, when inferring latent states from observations, we consider three different beliefs for each $\mathbf{z}_t$. Those are the prior $p(\mathbf{z}_t|\mathbf{o}_{\leq t-1}, \mathbf{a}_{\leq t-1})$, i.e., the belief before observing $\mathbf{o}_t$, the posterior, $p(\mathbf{z}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{\leq t-1})$, i.e., the belief after observing $\mathbf{o}_t$, as well as the smoothed belief $p(\mathbf{z}_t|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})$ which is conditioned on all future observations and actions, until the last time step $T$. We refer to those estimates as state-beliefs to distinguish them from dynamics distributions, i.e., distribu-



(a) State Space Model (b) *RSSM* Inference Model

Figure 1: (a) State space model, serving as the generative model throughout this work. (b) Graphical Model underlying the *RSSM*(Hafner et al., 2019) inference scheme. In contrast to the generative SSM, the direction between observations and latent states is inverted. These independence assumptions result in a simplified inference and subtle effects on model learning.

tions conditioned on the previous state $\mathbf{z}_{t-1}$ such as the prior dynamics $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1})$, the posterior dynamics $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t)$ and the smoothed dynamics $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_{\geq t})$. To get from a dynamics distribution to the corresponding state belief we need to marginalize out the previous state, i.e., $p(\mathbf{z}_t|\mathbf{o}_{\leq t-1}, \mathbf{a}_{\leq t-1}) = \int p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}) p(\mathbf{z}_{t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{\leq t-1}) d\mathbf{z}_{t-1}$ for the prior.

Traditionally, the independence assumptions of the generative model are also used for inference. Yet, *RSSMs* (Hafner et al., 2019) work with a different set of assumptions during inference, shown in Figure 1b[1]. In particular, in this graphical model the state $\mathbf{z}_t$ is conditionally independent of all observations $\mathbf{o}_{>t}$ and actions $\mathbf{a}_{\geq t}$ given $\mathbf{z}_{t-1}$, $\mathbf{a}_{t-1}$, and $\mathbf{o}_t$, which is not the case for the standard SSM. We discuss the effects of those assumptions on inference and model learning.

### 2.1 Inference in State Space Models

As exact inference is intractable for most models of interest, we usually use approximate methods such as variational inference. For a single sequence, the general variational lower bound decomposition to the log likelihood of the observations given the actions $\log p(\mathbf{o}_{\leq T}|\mathbf{a}_{\leq T})$ is given by

$$\log p(\mathbf{o}_{\leq T}|\mathbf{a}_{\leq T}) \geq \mathbb{E}_{q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}))} \left[\log p(\mathbf{o}_{\leq T}, \mathbf{z}_{\leq T}|\mathbf{a}_{\leq T}) - \log q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})\right]. \tag{1}$$

This bound is tight if $q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) = p(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})$. While this bound is valid for arbitrary distributions $q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})$, we need to pose a set of independence assumptions to obtain tractable inference models. If we use the same independence assumptions as the generative model, the inference model can be obtained by explicitly inverting the generative direction using Bayes rule. The resulting factorization can be read of the graphical model in Figure 1,

$$q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) = \frac{q(\mathbf{z}_{\leq T}, \mathbf{o}_{\leq T}|\mathbf{a}_{\leq T})}{q(\mathbf{o}_{\leq T}|\mathbf{a}_{\leq T})} = q(\mathbf{z}_0|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{\geq t-1}, \mathbf{o}_{\geq t}). \tag{2}$$

Inserting these into the general lower bound given in Equation 1 leads to $\mathcal{L}_{\text{ssm}}(\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) =$

$$\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{z}_t|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})} \left[\log p(\mathbf{o}_t|\mathbf{z}_t)\right] - \mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})} \left[\mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{\geq t-1}, \mathbf{o}_{\geq t}) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1})\right]\right]. \tag{3}$$

For certain parametrizations of the model, this variational distribution can be computed analytically, e.g., by Kalman smoothing if the model is linear and Gaussian, and in this case the bound is tight.

Yet, *RSSMs*, as introduced in (Hafner et al., 2019), assume the variational distribution factorizes as

$$q(\mathbf{z}_{\leq T}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) = q(\mathbf{z}_0|\mathbf{o}_0) \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t).$$

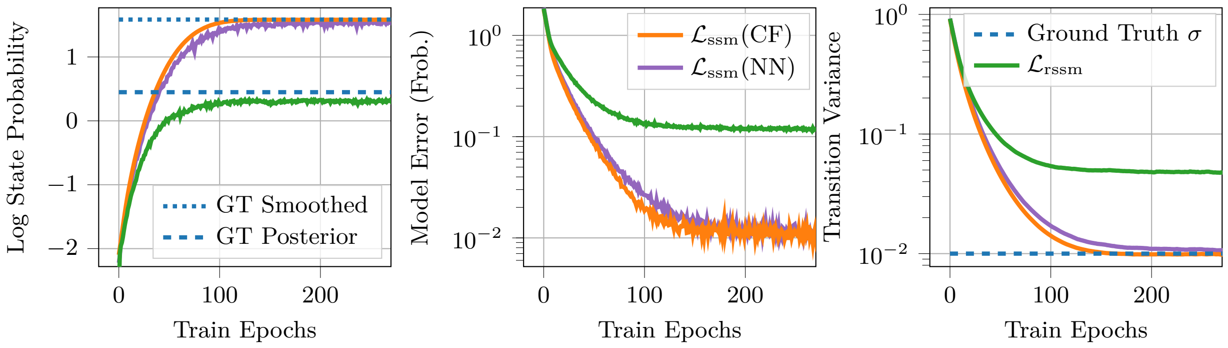
This assumption results in a simplified inference procedure, as the belief over $\mathbf{z}_t$ is assumed to be independent of all future observations $\mathbf{o}_{>t}$, given $\mathbf{z}_{t-1}$, $\mathbf{o}_t$, and $\mathbf{a}_{t-1}$. Inserting this assumptions into the general lower bound (Equation 1) gives the *RSSM*-bound introduced by Hafner et al. (2019), $\mathcal{L}_{\text{rssm}}(\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T}) =$

$$\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{z}_t|\mathbf{o}_{\leq t}, \mathbf{a}_{\leq t})} \left[\log p(\mathbf{o}_t|\mathbf{z}_t)\right] - \mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{\leq t-1})} \left[\mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1})\right]\right]. \tag{4}$$

This bound can not be tight, not even for linear Gaussian models, as the variational distribution does not consider future observations. Therefore, the resulting variational distribution is always different from $p(\mathbf{z}_{\leq t}|\mathbf{o}_{\leq T}, \mathbf{a}_{\leq T})$. Typically, tight variational bounds are preferable as they allow for faster optimization of the marginal log-likelihood. Yet, this discussion is more hypothetical, as all considered architectures do not provide tight lower bounds due to the use of deep neural networks, which prevents analytic solutions for the inference. However, as a tight lower bound does not even theoretically exist for the *RSSM* assumptions, we believe this is already an indication of the misspecification of its inference distribution.

---

[1] The full model of Hafner et al. (2019) also includes a *deterministic-path* which is of no concern regarding the discussion here. Thus, we omit it for brevity. We elaborate on this *deterministic-path* in Section 2.3. Further, Hafner et al. (2019) compare their *RSSM* to a baseline they abbreviate as *SSM* (stochastic state model), which also builds on the simplified assumptions. In this work, we refer to all approaches based on the simplified assumptions as *RSSM*.



(a) Inference Tracking Error (b) Model Learning Error (c) Transition Variance

Figure 2: Comparison of inference and model learning results under *RSSM* and SSM inference assumptions. We consider both a closed-form (CF) version based on Kalman Smoothing and a version with a neural network (NN) as an inference model for the SSM-based approaches and find that only the objective matters, not the parametrization of the inference model. **(a)** Log-probability of the ground truth states under the learned models. We compare against the quality of the ground truth smoothed (GT Smoothed) and posterior (GT Posterior) beliefs, computed using a Kalman Smoother and ground truth generative model. While the SSM objective reaches the quality of the smoothed belief, the *RSSM*-based inference fails to even attain the quality of the posterior belief. **(b)**: Distance between ground truth transition matrix and learned transition matrix, measured using the Frobenius norm. Here, the SSM inference yields a model that is an order magnitude closer to the ground-truth model than that learned by the *RSSM*-bound. **(c)**: Transition variance $\tilde{\sigma}\mathbf{I}$. With the SSM-bound we recover the ground-truth aleatoric uncertainty while with the *RSSM* bound the aleatoric uncertainty is significantly overestimated.

## 2.2 The Effects of Inference Assumptions on Model Learning

In the model-based RL setting considered in this work, we jointly learn the generative and inference models using an auto-encoding variational Bayes approach (Kingma & Welling, 2013; Sohn et al., 2015). Using an inference model that assumes future information cannot change the belief over a state has subtle effects on generative model learning. Indeed, as the belief over past states can by definition not change due to additional observations, any discrepancy between this past belief and the current observation must be explained by the transition model. In contrast, a smoothing inference can also explain the discrepancy by propagating information from observations to past beliefs. In Equation 4, this observation is reflected in the expected KL-term, $\mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{\leq t-1})}\left[\mathrm{KL}\left[q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t) \parallel p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{a}_{t-1})\right]\right]$. Here, the expectation does not consider $\mathbf{o}_t$ or other future observations, thus the transition model has to explain the transitions from a given $\mathbf{z}_{t-1}$ to $\mathbf{z}_t$, even if $\mathbf{z}_{t-1}$ would be rendered implausible by a future observation.

For a thought experiment illustrating these effects consider the following scenario. You meet a person holding a box and they tell you there is a hamster inside. As your prior experience is that people are usually trustworthy, you chose to believe them. Next, the box opens, and a cat jumps out. As you trust your eyes, you now believe it is a cat. Yet, under the *RSSM*-assumptions, you cannot revise your belief of the first time step and thus still believe it originally was a hamster. When updating your model based on this interaction, you would learn that hamsters can turn into cats, as you cannot capture the, arguably, more likely explanation that the person lied. Thus, learning under these assumptions requires you to model unlikely events as more likely than they are and overestimate the aleatoric uncertainty in the world.

More formally, we can demonstrate this effect using a simple linear-Gaussian State Space Model without actions. We will use a state dimension of 4 and the ground-truth generative model is given by

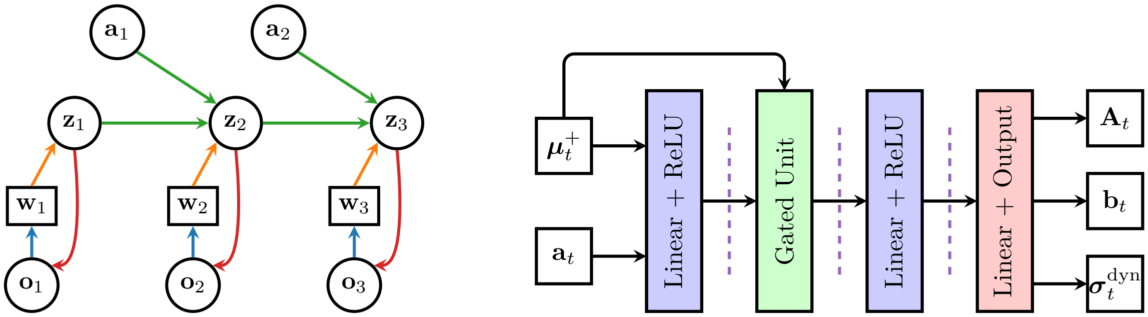
$$p(\mathbf{z}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad p(\mathbf{o}_t|\mathbf{z}_t) = \mathcal{N}(\mathbf{I}\mathbf{z}_t, 0.025\mathbf{I}), \quad p(\mathbf{z}_{t+1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{A}\mathbf{z}_t, 0.01\mathbf{I})$$

where $\mathbf{I}$ denotes the identity matrix. The transition matrix $\mathbf{A}$ induces a slightly damped, oscillating behavior. The complete matrix $\mathbf{A}$, together with further details regarding the exact setup of this experiment, can be found in Appendix B. Using this generative model, we generate $1,000$ sequences of length 50. Even in this simple setting, computing the optimal inference distribution for the *RSSM*-bound (Equation 4) is impossible, and we thus resort to numerical methods. We parameterize $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{o}_t)$ as locally linear-Gaussian distributions and learn their parameters using a neural network. For the SSM-bound (Equation 3 we can either compute the optimal inference in closed form, or again parameterize $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{o}_{\geq t})$ as a neural network. To condition on future observations, we use a GRU (Cho et al., 2014) which runs backward over the observation sequence. For the generative model, we learn a transition matrix $\tilde{\mathbf{A}}$ and an isotropic covariance $\tilde{\sigma}\mathbf{I}$ jointly with the parameters of the inference model. We fix the remaining parts of the generative model to the ground-truth values. Figure 2 summarizes the results and demonstrates that the *RSSM*-bound leads to a sub-optimal inference and consequently to learning a wrong model. In particular, we can see that for the *RSSM*, the transition variance $\tilde{\sigma}$ is much larger than the ground-truth value $\sigma = 0.01$ as all unexpected observations have to be explained by the transitions instead of correcting the beliefs of past time steps.

### 2.3 The Interplay of Policy Optimization and Regularization

Despite the theoretical considerations in the previous section, *RSSMs* work well for model-based RL. It is well known that model-based RL suffers from an *objective mismatch* (Luo et al., 2019; Lambert et al., 2020) issue. This issue arises because the model aims to maximize the ELBO (Equation 1) but is evaluated based on the agent's reward. The effect is further amplified by the distribution shift between training and data collection, as data collection is typically performed only after a policy improvement step. In RL, we explicitly want the agent to explore unseen parts of the state-action space, encountering observations the model has not seen before. Thus, training the underlying model requires careful regularization such that wrong predictions do not prevent the agent from exploring relevant parts of the state space. Many model-based RL approaches (Chua et al., 2018; Janner et al., 2019) handle this issue by explicitly modeling the epistemic uncertainty of the model, which is not required by the *RSSM*. Instead, we argue that *RSSMs* rely on the overestimated aleatoric uncertainty caused by sub-optimal inference to address *objective mismatch* in a more heuristic manner. The overestimation implicitly regularizes the *RSSM* as it forces the transition model to model unlikely events with higher probability. This regularization thus implicitly prevents overconfident model predictions due to overfitting. Yet, while it alleviates the *objective mismatch* issue, there are also drawbacks to this heuristic solution. First, it complicates the model design, analysis, and improvement of *RSSMs*. As already observed by Hafner et al. (2019), a fully stochastic model based on the *RSSM*-assumptions underperforms without additional measures as it fails at reliably propagating information for multiple time steps. As a remedy, Hafner et al. (2019) introduce a *deterministic-path*, i.e., a Gated Recurrent Unit (Cho et al., 2014), and base the belief update on this instead of the stochastic belief. Second, as we show in Section 4.1, there are settings where appropriately capturing the aleatoric uncertainty is important, and failing to do so can hurt performance.

In a first attempt to address those issues, we minimally adapt the *RSSM* to be capable of smoothing. This Smoothing *RSSM* uses a GRU which is added before the actual *RSSM* and runs backwards over the representations extracted by the encoder, effectively accumulating all future information. The *RSSM* then receives the output of this GRU instead of the original observation as input. When evaluating this model in Section 4.1, we find that the performance decreases compared to the original *RSSM*. We argue that with a proper inference the model can no longer rely on the overestimated aleatoric uncertainty for regularization and thus becomes more prone to *objective mismatch*. Following other methods, we try to improve the results by modeling epistemic uncertainty. To this end, we use Monte Carlo Dropout (MCD) but find that it does not help to improve the Smoothing *RSSM's* performance. From these results, we conclude that solely addressing the sub-optimal inference assumption is insufficient, but we also need to rethink the model's parameterization. We postulate that the additional GRU for the backward pass is a poor inductive bias and that we require a smoothing approach that adds as little complexity as possible to the model. In the next section, we will introduce an architecture that allows for parameter-free smoothing using a locally linear state space model in a latent space.



(a) *VRKN* Overview

(b) *VRKN* Dynamics Model Overview

Figure 3: **(a)** We use an encoder (blue) to extract a latent observation $\mathbf{w}_t$ and an uncertainty estimate from the observation $\mathbf{o}_t$. This latent observation is then used to update the state estimate $\mathbf{z}_t$ using the closed-form Kalman update (orange). We propagate the latent state estimate to the next time step using the transition model detailed in (b). A decoder (red) reconstructs the observation. **(b)** The transition model is a feed forward network that takes the current posterior mean $\boldsymbol{\mu}_t^+$ and the action $\mathbf{a}_t$ as input and emits the transition matrix $\mathbf{A}_t$, the offset term $\mathbf{b}_t$, and the transition noise $\boldsymbol{\sigma}_t^{\mathrm{dyn}}$. We find that a gated unit stabilizes the training compared to a simple feed-forward network. For a Bayesian treatment of the transition model's parameters, we include Monte Carlo Dropout layers at the positions indicated by the purple dashed lines.

## 3 Variational Recurrent Kalman Networks

To provide a theoretically better-grounded alternative to the *RSSM*, we require a model which allows tractable inference while still scaling to complex image-based control tasks. Further, the architecture should allow efficient computation of smoothed and posterior state beliefs and dynamics. While we need smoothed distributions for training, we require (filtered) posteriors for online control. Both should be computable by the same network architecture to avoid an over-parametrization and overfitting as a consequence. In addition, such an architecture can naturally integrate multiple sensors that emit data at different frequencies into a common latent state representation. To meet these criteria, we introduce a new parametrization of the latent dynamics based on a linear-Gaussian state space model (LGSSM)(Murphy, 2012) embedded in a latent space. The linear-Gaussian assumptions allow for efficient inference and rigorous treatment of uncertainties while working in a learned latent space allows for modeling high-dimensional and non-linear systems. We use a Bayesian treatment of the LGSSM's transition model by Monte Carlo Dropout (Gal & Ghahramani, 2016) to include epistemic uncertainty in our approach. We name the resulting approach *Variational Recurrent Kalman Network (VRKN)*. Figure 3 shows a schematic overview.

### 3.1 Learning the Latent Space.

To allow working with high-dimensional observations which depend non-linearly on the latent state, we introduce auxiliary, latent observations $\mathbf{w}_t$ for the original observations $\mathbf{o}_t$. This intermediate representation allows us to capture the highly complex relations between state and observations in the mapping from $\mathbf{o}_t$ to $\mathbf{w}_t$ (encoder) while using a simple, tractable mapping between the state and $\mathbf{w}_t$. We assume these latent observations are a deterministic encoding of the original observations, parameterized by a neural network. Following (Haarnoja et al., 2016; Becker et al., 2019), we extend this network with a second output, emitting uncertainty estimates $\boldsymbol{\sigma}_t^w$, i.e., strictly positive vectors of the same size as the latent observation. We provide the latent observations and the corresponding uncertainty estimates as input to the state space model described below. Thus, we assume that we obtain a latent observation sample from the encoder, not the mean of a distribution over latent observations. Intuitively, the variance encoder has to estimate the information content of the observation. For example, when estimating the latent observation uncertainty from images, some images might contain certain information, e.g., the positions of an object, while others do not. The former case would result in a low uncertainty and the latter in a high one. Note that this approach differs from many previous variational approaches, which model the latent observation as a random variable

(Watter et al., 2015; Karl et al., 2016; Fraccaro et al., 2017) which results in a more complex model and objective due to the additional latent variable. We assume a Gaussian generative distribution $p(\mathbf{o}_t|\mathbf{z}_t)$ with fixed variance and parameterize the mean by a neural network (decoder).

### 3.2 Latent Linear-Gaussian State Space Model.

**Observation Model.** Given the latent observations and uncertainty estimates from the encoder, we can assume a simple linear latent observation model for the SSM with $p(\mathbf{w}_t|\mathbf{z}_t) = \mathcal{N}\big(\mathbf{I}\mathbf{z}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^w)\big)$, i.e., the latent observation $\mathbf{w}_t$ is a noisy variant of the latent state $\mathbf{z}_t$ where the noise is given by the encoder's uncertainty estimates. Using this approach, we can employ the standard Kalman update rule to update the beliefs based on the current observation (Becker et al., 2019).

**Dynamics Model.** We model the latent dynamics as

$$p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{a}_t) = \mathcal{N}\left(\mathbf{A}_t(\boldsymbol{\mu}_t^+, \mathbf{a}_t)\mathbf{z}_t + \mathbf{b}_t(\boldsymbol{\mu}_t^+, \mathbf{a}_t), \boldsymbol{\sigma}_t^{\mathrm{dyn}}(\boldsymbol{\mu}_t^+, \mathbf{a}_t)\right),$$

where $\boldsymbol{\mu}_t^+$ denotes the mean of the posterior state estimate $p(\mathbf{z_t}|\mathbf{o}_{\leq t}, \mathbf{a}_{\leq t})$, i.e., we learn a linearized model around the current posterior mean. This linearization is identical to the one used in extended Kalman filtering (Jazwinski, 1970), where a known model is linearized around the current posterior mean. In particular, this model is linear in $\mathbf{z}_t$, enabling closed-form propagation of state beliefs.

We model $\mathbf{A}_t$ to be a diagonal matrix which is emitted together with the offset term $\mathbf{b}_t$ and the transition noise $\boldsymbol{\sigma}_t^{\mathrm{dyn}}$ by a single neural network $\phi^{\mathrm{dyn}}(\boldsymbol{\mu}_t^+, \mathbf{a}_t) = \left(\mathbf{A}_t, \mathbf{b}_t, \boldsymbol{\sigma}_t^{\mathrm{dyn}}\right)$ and carefully design this network to prevent the state estimates and gradients from growing indefinitely during training. First, as $\mathbf{A}_t$ is diagonal, its values are its eigenvalues, and constraining them in an appropriate range ensures stable dynamics. To this end, we use an activation of the form $f(x) = s \cdot \mathrm{sigmoid}(x + b) + m$ where we choose $s$, $b$, and $m$ such that it saturates at 0.1 and 0.99 while $f(0) = 0.9$. Here, the intuition is that we want plausible and stable dynamics, which we initialize as a slightly dampened system. Second, empirically, it is beneficial to not model $\phi^{\mathrm{dyn}}$ as a simple feed-forward network but, inspired by standard recurrent architectures (Hochreiter & Schmidhuber, 1997; Cho et al., 2014), employ a gating mechanism to mitigate problems with vanishing and exploding gradients. To this end, we use a standard GRU cell implementation but feed the posterior mean $\boldsymbol{\mu}_t^+$ into the memory input, i.e., $\phi^{\mathrm{dyn}}(\boldsymbol{\mu}_t^+, \mathbf{a}_t) = \phi_2(\mathrm{GRU}(\phi_1(\boldsymbol{\mu}_t^+, \mathbf{a}_t), \boldsymbol{\mu}_t^+))$. The resulting model is still fully stochastic and linear in $\mathbf{z}_t$. In contrast to the *RSSM*, it does not use a *determinstic-path* as the GRU cell does not introduce an additional deterministic memory and is used solely for addressing problems arising from unstable dynamics and gradients. Figure 3b provides an overview of the transition architecture.

**Initial State Distribution.** For the initial state distribution $p(\mathbf{z}_0)$, we use a Gaussian with zero mean and a learned diagonal variance which we initialize with the identity matrix.

**Sensor Fusion.** Given the possibility of using the Kalman update for incorporating observations, we can use the *VRKN* for a simple but principled approach to sensor fusion. Formally, we assume the observation $\mathbf{o}_t$ factorizes into $K$ different observations $\mathbf{o}_t^{(k)}$, i.e., $p(\mathbf{o}_t|\mathbf{z}_t) = \prod_{k=1}^K p(\mathbf{o}_t^{(k)}|\mathbf{z}_t)$. Those observations can be of various modalities and be available at different frequencies, e.g., high-frequency velocity information from an internal IMU and low-frequency camera images of the surroundings. In this scenario, we have $K$ encoders, one for each $\mathbf{o}_t^{(k)}$, and accumulate the latent observations by repeatedly applying the Kalman update. As the Kalman update is a simple instance of Bayesian conditioning, this architecture reflects the invariance to permutations of all observations for a single time-step. It also allows simply omitting the update if some of the $K$ observations are unavailable for a time step. Additionally, we have $K$ decoders and $K$ reconstruction loss terms in the ELBO.

### 3.3 Modeling Epistemic Uncertainty

As discussed in Section 2.3, the overestimated aleatoric uncertainty of the *RSSM*'s transition model avoids overconfident estimates due to overfitting and compensates for the lack of explicit epistemic uncertainty. Thus, as our approach captures the aleatoric uncertainty correctly, we need to explicitly consider epistemic uncertainty to obtain a model that is also useable for policy optimization. We use Monte Carlo Dropout

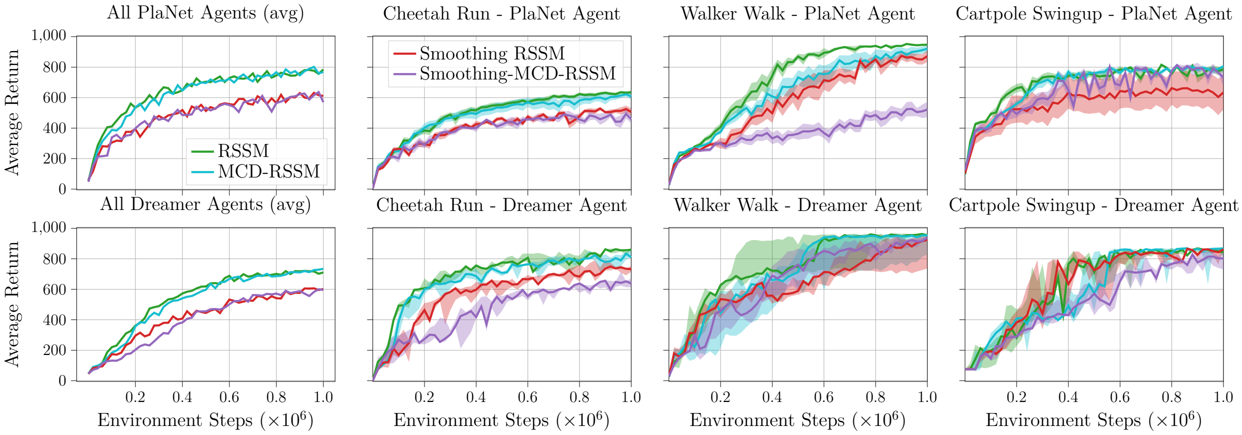


Figure 4: Comparison of the *RSSM*, the simple smoothing extension (Smoothing-*RSSM*) as well as versions of both models using Monte Carlo Dropout (MCD) to capture epistemic uncertainty in the dynamics (MCD-*RSSM* and Smoothing MCD-*RSSM*). The leftmost plot in each row shows the average performance of all considered agents. Those are not directly comparable as we use different environments for *PlaNet* and *Dreamer*-based agents. We find that proper inference by smoothing deteriorates performance on average, and the additional epistemic uncertainty does not compensate for this decrease in performance.

(Gal & Ghahramani, 2016) due to its simplicity and include corresponding layers at appropriate points in the transition model, see Figure 3b.

### 3.4 Inference and Training

To infer belief states using our model we first need to embed the observations in the latent space using the encoder. Given its output, i.e., the latent observations and uncertainty estimates, as well as the locally linear observation and dynamics model, we can compute the prior and posterior beliefs using the standard Kalman filter equations and smooth by iterating backward over those beliefs using the Rauch-Tung-Striebel (RTS) (Rauch et al., 1965) equations. Note that due to factorization induced by the diagonal transition matrices and observation covariance matrices, all Kalman updates can be reduced to scalar divisions instead of using costly matrix inversions (Becker et al., 2019). We train all parts of the model jointly by maximizing Equation 3, for which we need the smoothed dynamics $q(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{a}_{\geq t}, \mathbf{o}_{\geq t+1})$. We provide a recursive formula, extending the RTS equations, for computing these smoothed dynamics with minimal overhead in Appendix A.

When using the model for online control, we cannot smooth but have to act based on samples from the posterior estimate $q(\mathbf{z}_t|\mathbf{a}_{\leq t}, \mathbf{o}_{\leq t})$. Building on a latent LGSSM provides a strong inductive bias which prevents the model from learning a solution that relies on the backward recursion as it does not introduce new parameters and only uses quantities computed during the forward pass. Thus, the inductive bias allows learning reasonable posterior estimates without them explicitly being part of the training objective.

## 4 Evaluation

We compare the original *RSSM*, the smoothing *RSSM* and the *VRKN* on image-based continuous control tasks using the DeepMind Control Suite (Tassa et al., 2020). Prior works (Lambert et al., 2020; Lutter et al., 2021) concluded that the model's predictive performance is often uninformative about the quality of the model-based agent. We concluded the same after preliminary experiments and want to study the effects of the different assumptions and parametrizations on the performance in a model-based RL setting. Thus, we evaluate the state space models as backbones for model-based agents and directly consider the achieved reward. We use both the *PlaNet*(Hafner et al., 2019) and the *Dreamer*(Hafner et al., 2020) approaches for control. The *PlaNet*-agents plan actions using the cross entropy method by rolling out trajectories on the model. The *Dreamer*-agents learn a parametric policy and value function, using the model as a

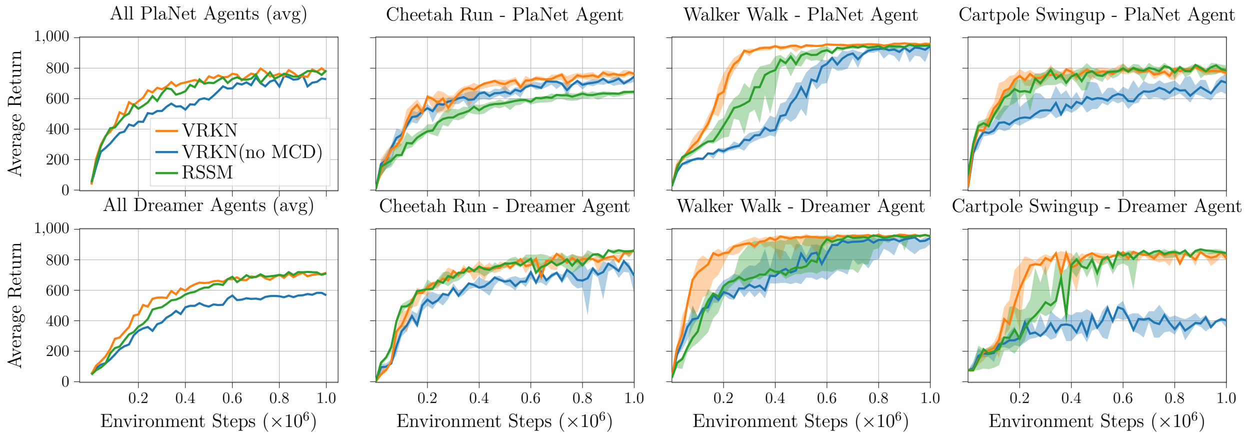


Figure 5: Comparison of *VRKN* and *RSSM* based *PlaNet* and *Dreamer* agents on the standard benchmarks. The leftmost plot in each row shows the average performance of all considered agents. Those are not directly comparable as we use different environments for *PlaNet* and *Dreamer*-based agents. While the *VRKN* without epistemic uncertainty (*VRKN (no MCD)*) cannot compete with the *RSSM*-agents, the Bayesian treatment allows it to reach a similar performance. In all environments, the *VRKN* profits from the epistemic uncertainty, either in terms of sample complexity, final performance, or both. Yet, for *Dreamer*-agents the Bayesian treatment is more relevant than in the *PlaNet* case. Further, for *Dreamer*-agents, the *VRKN* improves sample complexity in several environments.

differentiable simulator. The experiment setup closely follows the original works introducing *PlaNet* and *Dreamer*. Appendix C gives further details about the experimental setup and used baselines. We report the mean reward over all environments to show general trends and reward curves for the individual environments, showing median performance, where the shaded areas indicate 25% to 75%-percentiles. We use 5 seeds for each agent-environment pair, train for 1 million environment steps, and use 10 rollouts for evaluation. Appendix D provides further quantitative results and the reward curves for all considered environments.

### 4.1 Evaluation of the Effect of Epistemic Uncertainty on Different Smoothing Architectures

We start our evaluation by comparing the original *RSSM* with its extended smoothing version using a GRU with and without Monte Carlo Dropout (MCD), described in Section 2.3. For completeness, we also include a version of the original-*RSSM* with MCD to model epistemic uncertainty. For the *PlaNet*-agents, we evaluate the 6 environments originally used in (Hafner et al., 2019), for the *Dreamer*-agents we use 8 environments, i.e., Cheetah Run, Walker Walk, Cartpole Swingup, Cup Catch, Reacher Easy, Hopper Hop, Pendulum Swingup, and Walker Run. The proper smoothing inference deteriorates performance, as the model now lacks regularization to cope with the *objective mismatch*. Adding epistemic uncertainty in the form of MCD does, on average, neither affect the performance of the original *RSSM* nor the smoothing *RSSM*. Next, we compare the *VRKN* to the *RSSM* using the same environments and also include a version of the *VRKN* without Monte Carlo Dropout (MCD), i.e., without epistemic uncertainty, dubbed *VRKN (no MCD)*. Figure 5 shows a summary of the results and reward curves for some of the environments. Comprehensive results can be found in Appendix D.2. The results show that the *VRKN* relies on the epistemic uncertainty to match the *RSSMs* performance while *VRKN (no MCD)* cannot compete with the original *RSSM*-agents. In the case of the *Dreamer*-agents, we even find the *VRKN* converges faster to the final performance in several environments. These results emphasize regularization, either implicitly by sub-optimal inference or explicitly by capturing epistemic uncertainty, is important for model-based RL. Additionally, they indicate that epistemic uncertainty alone is insufficient for approaches using a correct smoothing inference. Those also rely on an appropriate inductive bias, as provided by the *VRKN*.

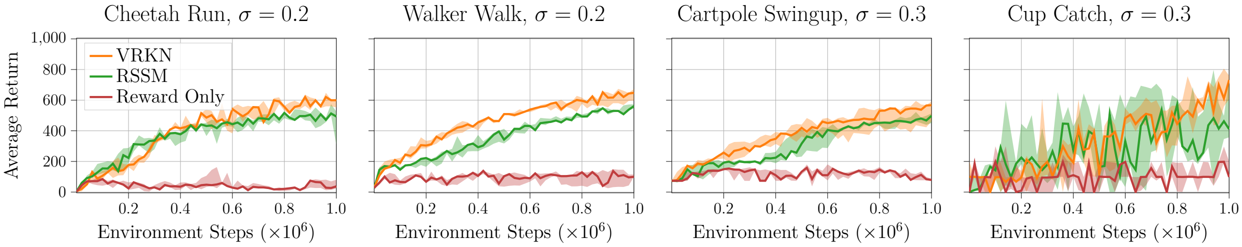

Cheetah Run, $\sigma = 0.2$ Walker Walk, $\sigma = 0.2$ Cartpole Swingup, $\sigma = 0.3$ Cup Catch, $\sigma = 0.3$

Figure 6: *Dreamer*-agent results on the occlusion task. The $\sigma$ in the title indicates the standard deviation of the Gaussian transition noise. In all tasks, the *VRKN* achieves better performance after 1 million environment steps. The comparison to the reward-only baseline, i.e., a *RSSM*-based agent trained without observation reconstruction, indicates that the occluded images still contain relevant information.

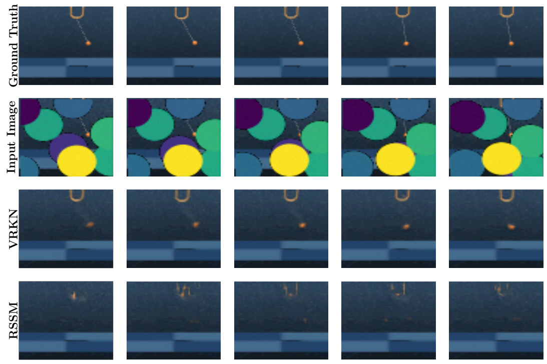


Figure 7: Exemplary sub-sequence of reconstructions, based on the model's posterior beliefs. The first row is the noise-free ground truth image, which the models never see. The second row is the model input, followed by the *VRKN* and the *RSSM* reconstructions. Even though the ball is partly visible in most images, the *RSSM* fails to reconstruct its position. The *VRKN* manages to do so and even provides a reasonable estimate for cup position. These results indicate the *VRKN's* improved ability to capture the system state in noisy scenarios.

### 4.2 Dealing with Occlusions

To better analyze the approaches' capabilities to capture and handle uncertainty, we design tasks where this is more important than in the almost noise-free standard benchmarks. To this end, we modify the observations of the environments Cheetah Run, Walker Walk, Cartpole Swingup, and Cup Catch. First, we introduce transition noise by adding Gaussian noise to the actions before execution. Second, we introduce additional observation uncertainty by rendering slow-moving occluding discs over the images. See the second row of Figure 7 for some examples. In the resulting environments, the prior beliefs are uncertain due to the transition noise, and not every observation has the same amount of useful information. Thus, the models need to correctly capture uncertainties in the system, allowing them to trade off information from the prior belief and current observation. We compare *Dreamer*-agents based on the *RSSM* and the *VRKN* and train using masked reconstruction, i.e., only non-occluded pixels contribute to the reconstruction loss[2]. We also consider a baseline where we train solely on the reward to show that the approaches can still extract information from the highly occluded observations. Figure 6 shows the results of the comparison. While the *RSSM's* and the *VRKN's* performance is almost identical in the standard versions of the considered environments (Figure 16), the *VRKN* works significantly better in the modified environments. Additionally, we qualitatively compare images reconstructed from posterior beliefs of both approaches to gain further insights into the quality of the belief state. Figure 7 shows some of those reconstructions for the Cup Catch task, further images can be found in Appendix D.5. From these images, it appears that the *VRKN* better captures the actual system state and uncertainty and thus allows the model-based agent to achieve a higher reward.

### 4.3 Dealing with Multiple Sensors at Different Frequencies

To test the models' capabilities for sensor fusion, we design a task where we provide only every $n$-th image, where $n$ is sampled uniformly between 4 and 8 while proprioceptive information is avail-

---

[2] We want to emphasize that we do not consider the availability of such loss masks a realistic scenario but see the task as a reasonable benchmark to evaluate the models' capabilities to cope with uncertainties.

able at every time step. The exact form of the proprioceptive information is task-dependent. For example, for Cup Catch, we define the cup position as proprioceptive, but not the ball position, which has to be inferred from images. Appendix C.3 provides an overview for all considered tasks. While for the *VRKN*, we can rely on the natural fusion mechanism described in Section 3, for the *RSSM*, we work with concatenation by feeding default values for unavailable observation and flags indicating whether the observation is valid. This experiment mimics a common robotics scenario where we have proprioceptive information about the robot at high frequencies but need to estimate the environment's state based on lower frequency images. It tests the approaches' abilities to form reasonable state estimates from observations that arrive in different modalities and at varying frequencies. Here again, the models need to trade off information encoded in the prior belief with the information available in both sensor sources, based on uncertainty. We test with and without transition noise and evaluate a baseline where we only provide the low-frequency images and no proprioceptive information. We consider the same 4 tasks as in the previous experiment and show average reuslts in Figure 8. For detailed results, we refer to the supplement. On average, the *VRKN* achieves a higher reward than the *RSSM*, especially in the setting with transition noise. This result again emphasizes the *VRKNs* capability to exploit all available information and appropriately capture the system's uncertainty.

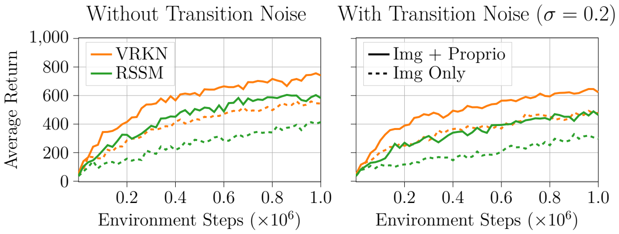


Figure 8: Average *Dreamer*-agent results over all considered environments on the fusion task. The $\sigma$ in the title indicates the standard deviation of the Gaussian transition noise. In general, the *VRKN* seems to exploit and accumulate the available information better, as the resulting agents perform better than those based on the *RSSM*. Further, the *VRKN* seems to cope better with the transition noise, most likely due to its appropriate modeling of the system's aleatoric uncertainty. Comprehensive results for individual environments can be found in Appendix D.3.

## 5 Related Work

***Recurrent State Space Models.*** While earlier works (Wahlström et al., 2015; Watter et al., 2015; Banijamali et al., 2018; Ha & Schmidhuber, 2018) showed the feasibility of control using learned latent state space models, the work originally proposing the *RSSM* (Hafner et al., 2019) was the first to show that such approaches can achieve similar performance to model-free RL on pixel-based complex continuous control tasks while using significantly fewer environment interactions. Since then, Hafner et al. (2020) improved their approach using a parametric policy learned on imagined trajectories and categorical latent spaces (Hafner et al., 2021). These approaches gained interest in the model-based RL community and are empirically successful, yet, little attention has been paid to the underlying state space model itself, the assumptions it builds upon, and its parametrization.

**State Space Models.** The Machine Learning community extensively studied and used state space models (SSMs). Besides classical approaches using linear models (Shumway & Stoffer, 1982) and works using Gaussian Processes (Eleftheriadis et al., 2017; Doerr et al., 2018), most recent methods build on Neural Networks (NNs). The first class of NN-based models of particular relevance for this work embeds linear-Gaussian SSMs (LGSSM) into latent spaces (Watter et al., 2015; Karl et al., 2016; Fraccaro et al., 2017; Banijamali et al., 2018; Becker-Ehmck et al., 2019; Klushyn et al., 2021). These approaches assume actuated systems and learn using stochastic gradient variational Bayes (Kingma & Welling, 2013). Yet, non of these approaches were used to model or even control systems of the complexity considered by (Hafner et al., 2019) and here. They are not directly applicable to these scenarios for various reasons. First, they use full transition matrices and covariances, which prevents them from scaling to sufficiently high-dimensional latent spaces. (Karl et al., 2016; Becker-Ehmck et al., 2019) do not allow smoothing. (Fraccaro et al., 2017; Klushyn et al., 2021) model the latent observations as random variables which are inferred jointly with the latent states and use constant observation uncertainty for the filtering in the latent space. This choice complicates inference and training and prevents principled usage of the observation uncertainty for filtering.

Our parameterization of the LGSSM alleviates these issues by building on factorization assumptions which yield a scalable architecture. Further, it allows smoothing and principled usage of observation uncertainty during filtering by modeling the observations in latent space as deterministic. Finally, non of these approaches considered modeling epistemic uncertainty.

Another class of approaches directly uses NN-based, nonlinear parametrization for SSMs (Archer et al., 2015; Krishnan et al., 2015; Gu et al., 2015; Zheng et al., 2017; Krishnan et al., 2017; Yingzhen & Mandt, 2018; Schmidt & Hofmann, 2018; Naesseth et al., 2018; Moretti et al., 2019). Out of this class, *Structured Inference Networks (SINs)* (Krishnan et al., 2017) are the most relevant for our work. *SINs* build on the same variational objective as *VRKN*, yet without conditioning on actions. The smoothing-*RSSM* baseline introduced in Section 2.3 can be considered an instance of a *SIN*. Yet while it builds on the same loss and fundamental ideas, the underlying NN architecture is very different.

**Kalman Updates in Deep Latent Space.** Haarnoja et al. (2016) first proposed using an encoder to extract uncertainty estimates from high-dimensional observations for filtering. They only learned the encoder while assuming the transition dynamics to be known. Becker et al. (2019) proposed an efficient factorization to additionally learn a high-dimensional, latent, locally-linear dynamics model. Shaj et al. (2020) extended this approach by introducing a principled form of action conditioning. While the *V-RKN* builds on many of their design choices, there are also considerable differences. Those mainly concern the parametrization of the dynamics model and further simplifying the factorization assumptions. These changes are necessary to make the approaches scale to the complex control tasks considered in this work. Additionally, Haarnoja et al. (2016); Becker et al. (2019); Shaj et al. (2020) train using regression and do not learn a full generative model. Thus, they cannot produce the reasonable latent trajectories needed for model-based RL.

**Epistemic Uncertainty for Model-Based RL.** Ample work emphasises the importance of modeling epistemic uncertainty for model-based RL (Deisenroth & Rasmussen, 2011; Chua et al., 2018; Janner et al., 2019) and several authors equipped *RSSMs* with epistemic uncertainty. Okada et al. (2020) use an ensemble of *RSSMs* and showed improved results on modified versions of the Deep Mind Control Suite (Tassa et al., 2020) benchmarks. Sekar et al. (2020) also combine an ensemble with the *RSSM* but focus on exploration and generalization to unseen tasks. Yet, neither of these works questioned the assumptions underlying the *RSSM* or analyzed their effects on the learned models.

## 6 Conclusion

We analyzed the independence assumptions underlying *Recurrent State Space Models (RSSMs)* and found they are theoretically sub-optimal. Yet, they implicitly regularize the model by causing an overestimated aleatoric uncertainty and are crucial to the *RSSMs* success in model-based RL. When trying to avoid this heuristic approach and use the correct assumptions while replacing the implicit regularization with a more explicit approach using epistemic uncertainty, we found a simple extension of the *RSSM* architecture is insufficient. Thus, we redesigned the model from first principles, using an inductive bias, which is appropriate for smoothing. As a result, we propose the *Variational Recurrent Kalman Network (VRKN)* which builds on well-understood tools for modeling the aleatoric and epistemic uncertainties. It uses extended Kalman smoothing for exact inference in a latent space to capture aleatoric uncertainty and explicit models the epistemic uncertainty using Monte Carlo Dropout. While agents based on the *VRKN* and the *RSSM* perform similar on the standard, noise-free benchmarks, the *VRKN*-agents significantly outperform those using the *RSSM* on tasks where capturing uncertainties is more relevant. Additionally, the *VRKN* provides a natural approach to sensor fusion and outperforms the RSSM on tasks that require fusing sensor observations from several sensors at different frequencies.

**Limitations.** We showed that designing a state space model out of well-founded components that at least matches the *RSSMs* performance is possible which opens a path to improve them individually. Yet, here we used simple instances of these components and have not yet further investigated how to improve them. Further, we have not investigated the interplay between the models and the controllers used on top of them but used the control approaches proposed in (Hafner et al., 2019) and (Hafner et al., 2020) with default parameters. Due to the intricate interplay between model learning and using the resulting controller for data collection, it is reasonable to rethink the design of the controller when changing the model.

## References

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. PMLR, 2018.

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, pp. 544–552, 2019.

Philip Becker-Ehmck, Jan Peters, and Patrick Van Der Smagt. Switching linear dynamics for variational Bayes filtering. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 553–562. PMLR, 09–15 Jun 2019.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

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.

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

Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Trimpe Sebastian. Probabilistic recurrent state-space models. In *International Conference on Machine Learning*, pp. 1280–1289. PMLR, 2018.

Stefanos Eleftheriadis, Tom Nicholson, Marc Peter Deisenroth, and James Hensman. Identification of gaussian process state space models. In *NIPS*, pp. 5309–5319, 2017.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, pp. 3601–3610, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

S Gu, Z Ghahramani, and RE Turner. Neural adaptive sequential monte carlo. *Advances in Neural Information Processing Systems*, 2015:2629–2637, 2015.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: learning discriminative deterministic state estimators. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4383–4391, 2016.

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*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=0oabwyZbOu`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32:12519–12530, 2019.

AH Jazwinski. *Stochastic processes and filtering theory.* ACADEMIC PRESS, INC.,, 1970.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models. *Advances in Neural Information Processing Systems*, 34, 2021.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *Proceedings of Machine Learning Research vol*, 120:1–15, 2020.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2019.

Michael Lutter, Leonard Hasenclever, Arunkumar Byravan, Gabriel Dulac-Arnold, Piotr Trochim, Nicolas Heess, Josh Merel, and Yuval Tassa. Learning dynamics models for model predictive agents. *arXiv preprint arXiv:2109.14311*, 2021.

Antonio Moretti, Zizhao Wang, Luhuan Wu, and Itsik Pe'er. Smoothing nonlinear variational objectives with sequential monte carlo, 2019.

Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pp. 968–977. PMLR, 2018.

Masashi Okada, Norio Kosaka, and Tadahiro Taniguchi. Planet of the bayesians: Reconsidering and improving deep planning network by incorporating bayesian inference. *arXiv preprint arXiv:2003.00370*, 2020.

Herbert E Rauch, F Tung, and Charlotte T Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

Florian Schmidt and Thomas Hofmann. Deep state space models for unconditional word generation. *Advances in Neural Information Processing Systems 31*, 31:6158–6168, 2018.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Vaisakh Shaj, Philipp Becker, Dieter Buchler, Harit Pandya, Niels van Duijkeren, C James Taylor, Marc Hanheide, and Gerhard Neumann. Action-conditional recurrent kalman networks for forward and inverse dynamics learning. *Conference on Robot Learning*, 2020.

Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control, 2020.

Niklas Wahlström, Thomas B Schön, and Marc Peter Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5670–5679. PMLR, 10–15 Jul 2018.

Xun Zheng, Manzil Zaheer, Amr Ahmed, Yuan Wang, Eric P Xing, and Alexander J Smola. State space lstm models with particle mcmc inference. *arXiv preprint arXiv:1711.11179*, 2017.