We thank the reviewers for their time and effort in providing helpful feedback. We address the comments in order.

Reviewer #1 – No report submitted.

Reviewer #2 *[Clarity]* Our supplementary material, Appendix D (ML Reproducibility), includes a detailed description of the network architectures used in our experiments. We used fully connected feedforward networks for both the nominal model $\hat{f}$ and the Lyapunov network V. It is worth noting, and we will add this to the paper, that for the nominal model $\hat{f}$ we can use any (parametric/differentiable – for training via backpropagation) representation of the form $x_{t+1} = \hat{f}(x_t, \omega_{t+1})$ to describe the state transitions. Therefore, other architectures such as convolutional neural networks or regularization techniques such as dropout are applicable. *[Relation to prior work]* We will elaborate more on the related literature section in the final paper. As for Umlauft and Hirche [39], their approach is constrained to quadratic Lyapunov functions. This constrains the state transition densities that their models can express. Our approach is more general, given the approximation capacity of mixture densities. Moreover, our method applies for non-convex Lyapunov networks and is readily applicable in both deterministic and stochastic settings.

Reviewer #3 *[Weaknesses]* We deliberately chose fairly complicated low-dimensional examples: as the reviewer says, these already have legitimate interest and allow for useful visualizations. But the method can readily be applied to high-dimensional states, as our model is a generic feedforward neural network. This allows for applications in model predictive control or reinforcement learning. *[Clarity]* We will add some notes on stability concepts and interpretations to the final paper. Here is a preview: 1) The first three definitions align well with deterministic definitions of stability and the corresponding Lyapunov theory, which is our starting point before deriving stability conditions in the stochastic setting; 2) The probabilistic definitions are very strong in the sense that they enforce certain boundedness conditions on the *random variable* $\sup_{t \geq 0} ||x_t||$, while stability in mean is a path-wise condition. We are then able to exploit properties about the mean/variance of a MDN as well as the Lyapunov network in order to unify Lyapunov theory with 2nd mean stability. In particular, we note that the (conditional) mean dynamics are themselves stochastic processes, which we constrain to be stable in the stochastic Lyapunov sense (there is some discussion of this at the end of section 4 – we force all sample conditional means to decrease in V). The 2nd mean condition specifically is simply convenient and could apply for any even-order moment by modifying the proof of Theorem 4.1 slightly. *[Additional feedback]* We will add variance info to our figures. Originally, the main drawback was simply how 'busy' the plots become when showing sample paths for both the true system and the learned model.

Reviewer #4 *[Weaknesses]* • We can appreciate the confusion that the abstract/introduction may have cause. We apologize for it. Let us clarify. "Deep dynamic models" can be used interchangeably with "DNN-based dynamic models" – DNNs for modeling a dynamical system. The purpose of the paper is to develop architectures with formal guarantees (stochastic stability) about such models. Stability is a critical property in real-world applications. This motivates our focus on stochastic discrete-time systems: even when the system of interest has continuous-time dynamics, we only observe samples $x_t, x_{t+1}, \ldots$ (rather than the functions $x(\cdot)$ and $\dot{x}(\cdot)$), making discrete-time stability the criterion of practical interest. Moreover, uncertainty (such as measurement noise) is unavoidable in practice, which is why we must account for stochasticity while pursuing stability. • We will expand on the discussion of our results in the final paper. Briefly, the first example unifies all the methods in the paper as well as the classical (deterministic) Lyapunov equation with a stochastic analog. The second illustrates how our implicit output layer method provides more flexibility than the convexity-based method for more complex dynamics. The third affirms the usefulness of stability as a 'primitive' in the model over a simple mixture density network. • From the points described in the first bullet, we cannot model a stochastic discrete system as a continuous deterministic system because the previous approach [30] is incompatible with discrete-time observations and does not address uncertainty. We will provide more details discussing the differences between our framework and the one in [30]. • In Appendices B (Training implicit dynamic mode) and D (ML Reproducibility) we provided a full description of the models, data, training, and gradient calculations for backpropagation (for the implicit layer). We will be more clear in the main text of the final paper about what is contained in the appendix. We will also include a GitHub link to our code in the final paper. *[Correctness]* Thank you for pointing this out. It is an important point that we will clarify. $\gamma^\star$ in Eq. (11) is a generalization of γ in Eq. (4) and is therefore state dependent as well. Eq. (10) expresses the problem for a fixed state, not all states simultaneously. *[Clarity]* • The first point is addressed above. • For the conceptual diagram, we felt Figure 1 gave a useful intuitive explanation of our two methods side-by-side. We will add a third sub-figure to illustrate stochastic component as well. • We will use Eq. (X) for clarity as suggested. *[Relation to prior work]* We address both of the reviewer's comments in the third bullet under *Weaknesses*. *[Reproducibility]* We address the reviewer's assessment in the fourth bullet under *Weaknesses*. *[Additional feedback]* This comment appears to be misplaced, but similar to the one in *Relation to prior work*. We have addressed it above, under *Weaknesses*.