# On the Stochastic Stability of Deep Markov Models

**Ján Drgoňa**[1], **Sayak Mukherjee**[1], **Jiaxin Zhang**[2], **Frank Liu**[2], **Mahantesh Halappanavar**[1]
[1] Pacific Northwest National Laboratory
Richland, Washington, USA
[2] Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
{jan.drgona, sayak.mukherjee, mahantesh.halappanavar}@pnnl.gov
{zhangj, liufy}@ornl.gov

## Abstract

Deep Markov models (DMM) are generative models that are scalable and expressive generalization of Markov models for representation, learning, and inference problems. However, the fundamental stochastic stability guarantees of such models have not been thoroughly investigated. In this paper, we provide sufficient conditions of DMM's stochastic stability as defined in the context of dynamical systems and propose a stability analysis method based on the contraction of probabilistic maps modeled by deep neural networks. We make connections between the spectral properties of neural network's weights and different types of used activation functions on the stability and overall dynamic behavior of DMMs with Gaussian distributions. Based on the theory, we propose a few practical methods for designing constrained DMMs with guaranteed stability. We empirically substantiate our theoretical results via intuitive numerical experiments using the proposed stability constraints.

## 1 Introduction

Modeling, analysis, and control of dynamical systems are of utmost importance for various physical and engineered systems such as fluid dynamics, oscillators, power grids, transportation networks, and autonomous driving, to name just a few. The systems are generally subjected to uncertainties arising from a plethora of factors such as exogenous noise, plant-model mismatch, and unmodeled system dynamics, which have led researchers to model the dynamics in stochastic frameworks. One of the most commonly used probabilistic frameworks to model dynamical system is the Hidden Markov Models (HMMs) [Rabiner and Juang, 1986, Eddy, 1996] which in their vanilla form have been extensively investigated for representation, learning, and inference problems [Ghahramani and Jordan, 1997, Cappé et al., 2006, Beal et al., 2002]. One of its variants, the Gaussian state space models, have been used in the systems and control community for decades [Beckers and Hirche, 2016, Eleftheriadis et al., 2017].

It has been shown that expressivity of Markov models to emulate complex dynamics and sequential behaviors is greatly improved by parametrizing such models using deep neural networks, giving rise to Deep Markov Models (DMMs) [Krishnan et al., 2017]. Main research activities have been focused on the inference of these models. In particular, works such as Awiszus and Rosenhahn [2018], Liu et al. [2019], Mustafa et al. [2019], Qu et al. [2019] proposed parametrizing the probability distributions using deep neural networks for modeling various complex dynamical systems. Despite the rising popularity of DMMs, many of their theoretical properties such as robustness to perturbations, and stability of the generated trajectories remain open research questions. Many natural systems exhibit complex yet stable dynamical behavior that is described by converging trajectories towards an attractor set [Brayton and Tong, 1979]. Additionally, safety-critical systems such as an autonomous

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

driving call for formal verification methods to ensure safe operation. Thus the ability to guarantee stability in data-driven models could lead to improved generalization or act as a safety certificate in real-world applications.

In this paper, we propose a new analytical method to assess stochastic stability of DMMs. More specifically, we utilize spectral analysis of deep neural networks (DNNs) modeling DMM's distributions. This allows us to make connections between the stability of deterministic DNNs and the stochastic stability of deep Markov models. As a main theoretical contribution we provide sufficient conditions for stochastic stability of DMMs. Based on the proposed theory we introduce several practical methods for design of constrained DMM with stability guarantees. In summary, the main contributions of this paper include:

1. **Stability analysis method for deep Markov models:** we base the analysis on the operator norms of deep neural networks modeling mean and variance of the DMM's distributions.
2. **Sufficient conditions for stochastic stability of deep Markov models:** we show the sufficiency of the operator norm-based contraction conditions for DMM's deep neural networks by leveraging Banach fixed point theorem.
3. **Stability constrained deep Markov models:** we introduce a set of methods for the design of provably stable deep Markov models.
4. **Numerical case studies:** we analyze connections between the design parameters of neural networks, stochastic stability, and operator norms of deep Markov models.

## 2 Related Work

Deep Markov models (DMMs) have been used as a scalable and expressive generalization of Hidden Markov Models (HMM) for learning probabilistic generative models of complex high-dimensional dynamical systems from sequential data [Rezende et al., 2014, Krishnan et al., 2017, Fraccaro et al., 2016]. DMMs have been successfully applied to speech recognition problems [Li et al., 2013, Prasetio et al.], control problems [Shashua and Mannor, 2017], human pose forecasting [Toyer et al., 2017], fault detection [Wang et al., 2018], climate data forecasting [Che et al., 2018], molecular dynamics [Wu et al., 2018], or as internal models in model-based deep RL applied to automatic trading [Ferreira, 2020]. Several modifications of DMMs have been proposed to handle incomplete data [Tan et al., 2019], or multi-rate time series [Che et al., 2018] multivariate time series [Montanez et al., 2015], training DMMs in unsupervised settings [Tran et al., 2016], or architectures inspired by Kalman Filters [Krishnan et al., 2015, Shashua and Mannor, 2017, Becker et al., 2019]. However, works focusing on formal analysis to ensure stability guarantees for DMMs are missing.

Stability notions and analysis for stochastic dynamic systems have been studied in the automatic control literature in various forms, depending on the representation of the system dynamics. Some classical results on stochastic stability for analysis and control can be found in McLane [1971], Willems and Willems [1976]. These results are presented mainly for stochastic differential equation (SDE) models. Khasminskii [2012] discusses different notions of stochastic stability, among them mean-square based stability notions have gained interest in works such as Lu and Skelton [2002], Farmer et al. [2009], Elia et al. [2013], Nandanoori et al. [2018], Wu et al. [2019]. We resort to such mean-square based stability notions when analyzing probabilistic state transition models parametrized by deep neural networks. We show that when the stochastic transitions are modeled by DNNs, the probabilistic stability requirements can be translated to deterministic stability notions of nonlinear discrete-time dynamics [Khalil, 2002].

In recent years, deep neural networks have been extensively studied from the viewpoint of dynamical systems [Chen et al., 2018, Raissi et al., 2017, Ciccone et al., 2018], allowing for the application of stability analysis methods to DNNs. For instance, Manek and Kolter [2019] proposed neural Lyapunov function to stabilize learned neural dynamics of autonomous system. Haber and Ruthotto [2017] interpret residual connections in neural networks as Euler discretization of ODEs and provide stability guarantees of ResNet. Goel and Klivans [2017] makes connections between eigenvalue decay and learnability of neural networks. Engelken et al. [2020], Vogt et al. [2020] studies the Lyapunov spectrum to of the input-output Jacobian of recurrent neural networks (RNNs) to assess RNN's stability. In this work we leverage advances in the spectral analysis of deep neural networks and apply them to derive stochastic stability guarantees for DMMs.

Besides analytical methods, many authors introduced provably stable neural architectures [Haber et al., 2019, Greydanus et al., 2019, Cranmer et al., 2020] or stability constraints [John et al., 2017]. Another popular strategy is to employ stabilizing regularizations. This can be achieved by minimizing eigenvalues of squared weights [Ludwig et al., 2014], using symplectic weights [Haber and Ruthotto, 2017], orthogonal parametrizations [Mhammedi et al., 2017], Perron-Frobenius theorem [Tuor et al., 2020], Gershgorin dics theorem [Lechner et al., 2020], or via singular value decomposition (SVD) [Zhang et al., 2018]. In this paper we leverage different weight factorization to empirically validate the proposed theoretical guarantees on stochastic stability of DMMs.

# 3 Methodology

This section presents stochastic stability analysis method for deep Markov models (DMM). First, we demonstrate the equivalence of DNNs with pointwise affine (PWA) functions. Next, we recall the definition of DMM with transition probabilities modeled by deep neural networks (DNNs). We introduce definitions of stochastic stability and show how can we leverage deterministic stability analysis in the probabilistic context. Finally, we will leverage the equivalence of DNNs with PWA maps to pose sufficient stability stochastic conditions for DMMs based on contraction of PWA maps.

## 3.1 Equivalence of Deep Neural Networks with Pointwise Affine Maps

Let us consider deep neural network (DNN) $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^m$ parametrized by $\theta_{\boldsymbol{\psi}} = \{\mathbf{A}_0^{\boldsymbol{\psi}}, \dots \mathbf{A}_L^{\boldsymbol{\psi}}, \mathbf{b}_0, \dots, \mathbf{b}_L\}$ with hidden layers $1 \leq l \leq L$ with bias given as follows:

$$\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(\mathbf{x}) = \mathbf{A}_L^{\boldsymbol{\psi}} \mathbf{h}_L^{\boldsymbol{\psi}} + \mathbf{b}_L \tag{1a}$$

$$\mathbf{h}_l^{\boldsymbol{\psi}} = \boldsymbol{v}(\mathbf{A}_{l-1}^{\boldsymbol{\psi}} \mathbf{h}_{l-1}^{\boldsymbol{\psi}} + \mathbf{b}_{l-1}) \tag{1b}$$

with $\mathbf{h}_0^{\boldsymbol{\psi}} = \mathbf{x}$, and $\boldsymbol{v} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$ representing element-wise application of an activation function to vector elements such that $\boldsymbol{v}(\mathbf{z}) := [\boldsymbol{v}(z_1) \dots \boldsymbol{v}(z_{n_z})]^\intercal$.

**Lemma 1.** *For a multi-layer feedforward neural network $\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}$ (3.2) with arbitrary activation function $\boldsymbol{v}$, there exists an equivalent pointwise affine map (PWA) parametrized by $\mathbf{x}$ which satisfies:*

$$\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(\mathbf{x}) = \mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x}). \tag{2}$$

*Where $\mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})$ is a parameter varying matrix given as:*

$$\mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} = \mathbf{A}_L^{\boldsymbol{\psi}} \boldsymbol{\Lambda}_{\mathbf{z}_{L-1}}^{\boldsymbol{\psi}} \mathbf{A}_{L-1}^{\boldsymbol{\psi}} \dots \boldsymbol{\Lambda}_{\mathbf{z}_0}^{\boldsymbol{\psi}} \mathbf{A}_0^{\boldsymbol{\psi}} \mathbf{x} \tag{3}$$

*And $\mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x})$ is a parameter varying vector, both parametrized by input vector $\mathbf{x}$ given by following recurrent formula:*

$$\mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x}) = \mathbf{b}_L^{\boldsymbol{\psi}} \tag{4}$$

$$\mathbf{b}_l^{\boldsymbol{\psi}} := \mathbf{A}_i^{\boldsymbol{\psi}} \boldsymbol{\Lambda}_{\mathbf{z}_{l-1}}^{\boldsymbol{\psi}} \mathbf{b}_{l-1}^{\boldsymbol{\psi}} + \mathbf{A}_i^{\boldsymbol{\psi}} \boldsymbol{\sigma}_{l-1}(\mathbf{0}) + \mathbf{b}_l, \;\; l \in \mathbb{N}_1^L \tag{5}$$

*with $\mathbf{b}_0^{\boldsymbol{\psi}} = \mathbf{b}_0$, and $i$ representing index of the network layer. Here $\boldsymbol{\Lambda}_{\mathbf{z}_l}^{\boldsymbol{\psi}}$ represents parameter varying diagonal matrix of activation patterns defined as:*

$$\boldsymbol{\sigma}(\mathbf{z}) = \begin{bmatrix} \frac{\sigma(z_1)-\sigma(0)}{z_1} & & \\ & \ddots & \\ & & \frac{\sigma(z_n)-\sigma(0)}{z_n} \end{bmatrix} \mathbf{z} + \begin{bmatrix} \sigma(0) \\ \vdots \\ \sigma(0) \end{bmatrix} = \boldsymbol{\Lambda}_{\mathbf{z}}^{\boldsymbol{\psi}} \mathbf{z} + \boldsymbol{\sigma}(\mathbf{0}) \tag{6}$$

*Proof.* First lets observe the following:

$$\boldsymbol{\sigma}(\mathbf{z}) = \begin{bmatrix} \sigma(z_1) \\ \vdots \\ \sigma(z_n) \end{bmatrix} = \begin{bmatrix} \frac{z_1(\sigma(z_1)-\sigma(0)+\sigma(0))}{z_1} \\ \vdots \\ \frac{z_n(\sigma(z_n)-\sigma(0)+\sigma(0))}{z_n} \end{bmatrix} = \begin{bmatrix} \frac{\sigma(z_1)-\sigma(0)}{z_1} & & \\ & \ddots & \\ & & \frac{\sigma(z_n)-\sigma(0)}{z_n} \end{bmatrix} \mathbf{z} + \begin{bmatrix} \sigma(0) \\ \vdots \\ \sigma(0) \end{bmatrix} \tag{7}$$

Remember $\sigma(0) - \sigma(0) = 0$, and $\frac{z_i}{z_i} = 1$ are identity elements of addition and multiplication, respectively. Thus (7) demonstrates the equivalence $\boldsymbol{\sigma}(\mathbf{z}) = \boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}}\mathbf{z} + \boldsymbol{\sigma}(\mathbf{0})$ as given in (6). Then if we let $\mathbf{z}_l = \mathbf{A}^{\boldsymbol{\psi}}_l\mathbf{x}_l + \mathbf{b}_l$, we can represent a neural network layer in a parameter varying affine form:

$$\boldsymbol{\sigma}_l(\mathbf{A}^{\boldsymbol{\psi}}_l\mathbf{x}_l + \mathbf{b}_l) = \boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_l}(\mathbf{A}^{\boldsymbol{\psi}}_l\mathbf{x}_l + \mathbf{b}_l) + \boldsymbol{\sigma}(\mathbf{0}) = \boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_l}\mathbf{A}^{\boldsymbol{\psi}}_l\mathbf{x}_l + \boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_l}\mathbf{b}_l + \boldsymbol{\sigma}_l(\mathbf{0}) \tag{8}$$

Now for simplicity of exposition lets assume only activations with trivial null space, i.e. $\sigma(0) = 0$. Thus $\boldsymbol{\sigma}(\mathbf{z}) = \boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}}\mathbf{z}$. By composition, a DNN $\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(\mathbf{x})$ can now be formulated as a parameter-varying affine map $\mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x})$, parametrized by input $\mathbf{x}$

$$\begin{aligned}
&\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(\mathbf{x}) := \mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x}) = \\
&\mathbf{A}^{\boldsymbol{\psi}}_L\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_{L-1}}(\mathbf{A}^{\boldsymbol{\psi}}_{L-1}(\ldots\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_1}(\mathbf{A}^{\boldsymbol{\psi}}_1\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_0}(\mathbf{A}^{\boldsymbol{\psi}}_0\mathbf{x} + \mathbf{b}_0) + \mathbf{b}_1)\ldots) + \mathbf{b}_{L-1})\mathbf{x} + \mathbf{b}_L \\
&\mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} = \mathbf{A}^{\boldsymbol{\psi}}_L\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_{L-1}}\mathbf{A}^{\boldsymbol{\psi}}_{L-1}\ldots\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_0}\mathbf{A}^{\boldsymbol{\psi}}_0\mathbf{x} \\
&\mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x}) = \mathbf{A}^{\boldsymbol{\psi}}_L\ldots\mathbf{A}^{\boldsymbol{\psi}}_2\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_1}\mathbf{A}^{\boldsymbol{\psi}}_1\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_0}\mathbf{b}_0 + \mathbf{A}^{\boldsymbol{\psi}}_L\ldots\mathbf{A}^{\boldsymbol{\psi}}_2\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_1}\mathbf{A}^{\boldsymbol{\psi}}_1\boldsymbol{\sigma}_0(\mathbf{0}) + \\
&\mathbf{A}^{\boldsymbol{\psi}}_L\ldots\mathbf{A}^{\boldsymbol{\psi}}_2\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_1}\mathbf{b}_1 + \mathbf{A}^{\boldsymbol{\psi}}_L\ldots\mathbf{A}^{\boldsymbol{\psi}}_2\boldsymbol{\sigma}_1(\mathbf{0}) + \ldots + \mathbf{A}^{\boldsymbol{\psi}}_L\boldsymbol{\Lambda}^{\boldsymbol{\psi}}_{\mathbf{z}_{L-1}}\mathbf{b}_{L-1} + \mathbf{A}^{\boldsymbol{\psi}}_L\boldsymbol{\sigma}_{L-1}(\mathbf{0}) + \mathbf{b}_L
\end{aligned} \tag{9}$$

Hence, each input feature vector $\mathbf{x}$ generates a unique affine map $\mathbf{A}_{\boldsymbol{\psi}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\boldsymbol{\psi}}(\mathbf{x})$ of the DNN $\boldsymbol{\psi}_{\theta_{\boldsymbol{\psi}}}(\mathbf{x})$. Thus proving the equivalence of DNN map (3.2) with the form (2). The case with $\boldsymbol{\sigma}(\mathbf{0}) \neq 0$ can be derived following the same algebraic operations as as above. □

### 3.2 Deep Markov Models

We consider a dynamical system with latent state variables $\mathbf{x}_t \in \mathbb{R}^n$, and the observed variables $\mathbf{y}_t \in \mathbb{R}^m$. The transition from $\mathbf{x}_t$ to the next time step $\mathbf{x}_{t+1}$, and the outputs $\mathbf{y}_t$ are modeled by probabilistic transitions. Over a horizon of $T$ time steps with a step size $\Delta t$, we assume the Markov property to embed structural independence conditions in the dynamic state evolution, i.e.,

$$\mathbf{x}_{t+1} \perp \mathbf{x}_{0:t-1} \mid \mathbf{x}_t, \tag{10}$$

Thus having latent state dynamics characterized by one-time-step conditional distribution $P(\mathbf{x}_{t+1}|\mathbf{x}_t)$. The joint distribution over the latent states and the observations is given by,

$$P(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}) = P(\mathbf{x}_0)P(\mathbf{y}_0|\mathbf{x}_0)\prod_{t=0}^{T-1}P(\mathbf{x}_{t+1}|\mathbf{x}_t)P(\mathbf{y}_t|\mathbf{x}_t). \tag{11}$$

More explicitly, we consider the following probabilistic transition and the emission maps:

$$\mathbf{x}_{t+1} \sim \mathcal{N}(K_\alpha(\mathbf{x}_t, \Delta t), L_\beta(\mathbf{x}_t, \Delta t)) \quad \text{(Transition)} \tag{12a}$$
$$\mathbf{y}_t \sim \mathcal{M}(F_\kappa(\mathbf{x}_t)) \quad \text{(Emission)} \tag{12b}$$

with the initial condition $\mathbf{x}_{t=0} = \mathbf{x}_0$. Here, $\mathcal{N}$ and $\mathcal{M}$ denote the probability distributions. For the transition mapping, $\mathcal{N}$ denotes a Gaussian distribution with the mean vector $K_\alpha(\mathbf{x}_t, \Delta t)$, and covariances $L_\beta(\mathbf{x}_t, \Delta t)$. The distribution $\mathcal{M}$ can be arbitrary with its distribution characterized by the map $F_\kappa(\mathbf{x}_t)$. In this paper, we are interested in the stability characterizations of the latent state dynamics given by transition map (12a), thereby assuming full state observability.

We are interested in expressing the dynamics of complex systems using (12), therefore it is suitable to expand the expressivity of our model by parametrizing the conditional distribution $P(\mathbf{x}_{t+1}|\mathbf{x}_t)$ by deep neural networks (DNNs) given as,

$$K_\alpha(\mathbf{x}_t, \Delta t) = \mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x}_t), \tag{13}$$
$$\text{vec}(L_\beta(\mathbf{x}_t, \Delta t)) = \mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x}_t), \tag{14}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n^2}$ are two deep neural networks parametrized by $\theta_\mathbf{f}$ and $\theta_\mathbf{g}$, respectively. And vec$(\cdot)$ denotes standard vectorization operation. Therefore, the probabilistic transition dynamics (12a) can be characterized by analysing the stability and boundedness of deep neural networks $\mathbf{f}$ and $\mathbf{g}$.

### 3.3 Stability of Deep Markov Models

To this end, we bring forth a few stability notions in the context of stochastic state transitions. In stochastic dynamics and control literature [Khasminskii, 2012], various different notions of stability for stochastic state transitions, such as mean-square stability, almost-sure stability and stability via convergence in probability, have been discussed. In this article, since we are interested in the latent state trajectories of the dynamic systems, we consider the mean-square stability as defined in [Willems and Willems, 1976, Nandanoori et al., 2018]. The dynamic system is said to be mean-square stable if the first and the second moment converge over time.

**Definition 2.** *The stochastic process $\mathbf{x}_t \in \mathbb{R}^n$ is mean-square stable (MSS) if and only if there exists $\mu \in \mathbb{R}^n, \mathbf{\Sigma} \in \mathbb{R}^{n\times n}$, such that $\lim_{t\to\infty} \mathbb{E}(\mathbf{x}_t) = \mu$, and $\lim_{t\to\infty} \mathbb{E}(\mathbf{x}_t\mathbf{x}_t^T) = \mathbf{\Sigma}$.*

The MSS condition from Definition 2 requires the dynamics (13) to have stable equilibrium $\bar{\mathbf{x}}_e = \mu$, where $\bar{\mathbf{x}}$ denotes the mean state vector. The definition also requires the covariance to converge. To express the dynamic behavior of complex system, the second moment convergence criterion can be relaxed by only requiring it to be norm bounded in order to ensure stochastic stability given as,

$$\|\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x}_t)\|_p < K,\ K > 0,\ \forall t. \tag{15}$$

Here $\|\cdot\|_p$ denotes any appropriate vector norm, e.g., $L2-$norm. The bound on the covariances will depend on the extent of stochasticity that the dynamic system encounters in an uncertain environment. However, in the later parts, we consider the convergence scenario for $\mathbf{g}$ as in Definition 2, rather than merely boundedness, in prescribing the sufficient conditions for the MSS-type stability.

Let us first analyze the mean dynamics characterized by (13), and its equilibrium $\bar{\mathbf{x}}_e = \mu$ which satisfies the stationarity condition $\mathbf{f}_{\theta_\mathbf{f}}(\bar{\mathbf{x}}_e) = \bar{\mathbf{x}}_e$. We have the mean state vector $\bar{\mathbf{x}}_t$ evolving under the following dynamics:

$$\bar{\mathbf{x}}_{t+1} = \mathbf{f}_{\theta_\mathbf{f}}(\bar{\mathbf{x}}_t). \tag{16}$$

(16) allows us to analyze the asymptotic stability of the DMM mean dynamics. Now in the main result of this section we leverage the fact that the dynamic characteristics of the deep neural networks $\mathbf{f}_{\theta_\mathbf{f}}, \mathbf{g}_{\theta_\mathbf{g}}$ around a point $\bar{\mathbf{x}}_t$ can be evaluated by obtaining their exact pointwise affine forms (PWA) (2). Based on this equivalence we formulate Theorem 3 and Corollary 5 as follows with sufficient conditions for the stability of deep Markov models.

**Theorem 3.** *The deep Markov model* (12) *which is parametrized by deep neural networks* (13)-(14) *remains globally stable in the mean-square sense if the following holds. The mean neural network $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ is a contractive map for any $\mathbf{x}$ in the domain of $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$. The variance network $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ is bounded for any $\mathbf{x}$ in the domain of $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$. Or more formally:*

$$\|\mathbf{A}_\mathbf{f}(\mathbf{x})\|_p < 1 \tag{17a}$$

$$||\mathbf{A}_\mathbf{g}(\mathbf{x})||_p + \frac{||\mathbf{b}_\mathbf{g}(\mathbf{x})||_p}{\|\mathbf{x}\|_p} < 1, \tag{17b}$$

$$\forall \mathbf{x} \in \textit{Domain}(\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x}), \mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})). \tag{17c}$$

*Proof.* First we prove the sufficiency of the contraction condition of the mean dynamics (17a). We base the proof on the equivalence of multi-layer neural networks with pointwise affine maps (2). An affine map is a contraction if the 2-norm of its linear part is bounded below one, i.e. $||\mathbf{A}||_2 < 1$. Thus it follows that the condition (17a) and equivalence (2) imply a contractive mean neural network $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$. The sufficiency of the contraction condition on mean square stable (MSS) equilibrium in the sense of Definition 2 follows directly from the Banach fixed-point theorem, which states that every contractive map converges towards single point equilibirum. Hence condition (17a) implies convergent mean transition dynamics:

$$\mu = \mathbf{f}_{\theta_\mathbf{f}}(\mu) = \lim_{t\to\infty} \mathbf{f}_{\theta_\mathbf{f}}(\bar{\mathbf{x}}_t) \tag{18}$$

Now we show the sufficiency of (17b) to guarantee the boundedness of the covariance matrix elements (15) by bounding the $p$-norm of the covariance neural network $\|\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})\|_p$. Please note that using the form (2) gives us $\|\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})\|_p = \|\mathbf{A}_\mathbf{g}(\mathbf{x})\mathbf{x} + \mathbf{b}_\mathbf{g}(\mathbf{x})\|_p$ yielding following inequalities:

$$\|\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})\|_p \leq \|\mathbf{A}_\mathbf{g}(\mathbf{x})\mathbf{x}\|_p + \|\mathbf{b}_\mathbf{g}(\mathbf{x})\|_p, \tag{19a}$$

$$\frac{||\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})||_p}{\|\mathbf{x}\|_p} \leq \|\mathbf{A}_\mathbf{g}(\mathbf{x})\|_p + \frac{||\mathbf{b}_\mathbf{g}(\mathbf{x})||_p}{\|\mathbf{x}\|_p}. \tag{19b}$$

We show that (19b) gives in fact a local Lipschitz constant of the variance network $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$. We exploit the point-wise affine nature of the neural network's form (2) and the fact that the norm of a linear operator $\mathbf{A}$ is equivalent to its minimal Lipschitz constant $\mathcal{K}^A_{min} = ||\mathbf{A}||_p$ [Huster et al., 2018]. Thus we can compute the local Lipschitz constants of a neural network $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ as:

$$\mathcal{K}^\mathbf{g}(\mathbf{x}) = ||\mathbf{A}_\mathbf{g}(\mathbf{x})||_p + \frac{||\mathbf{b}_\mathbf{g}(\mathbf{x})||_p}{\|\mathbf{x}\|_p}. \tag{20}$$

Applying the upper bound (17b) on the local Lipschitz constant (20) guarantees the contraction of the variance neural network $\mathbf{f}_{\theta_\mathbf{g}}(\mathbf{x})$ towards a fixed steady state $\mathbf{\Sigma}$. □

**Remark 4.** *To guarantee stochastic stability, the condition* (17b) *can be relaxed as given in* (15) *to bounded second moment* (15) *with* $\max_\mathbf{x} \mathcal{K}^\mathbf{g}(\mathbf{x}) < K$*, where* $K > 0$*.*

**Corollary 5.** *The deep Markov model* (12) *which is parametrized by deep neural networks* (13)-(14) *remains globally stable in the mean-square sense if the following holds: All weights* $\mathbf{A}^\mathbf{f}_i$ *of the mean network are* $\mathbf{f}_{\theta_\mathbf{f}}$ *contractive maps. All activation scaling matrices* $\mathbf{\Lambda}^\mathbf{f}_{\mathbf{z}_i}$ *of the mean network are non-expanding. Norms of all weights* $\mathbf{A}^\mathbf{g}_j$ *and activation scaling matrices* $\mathbf{\Lambda}^\mathbf{g}_{\mathbf{z}_j}$ *of the variance network* $\mathbf{g}_{\theta_\mathbf{g}}$ *are upper bounded by* 1*. Or more formally:*

$$\|\mathbf{A}^\mathbf{f}_i\|_p < 1,\ ||\mathbf{\Lambda}^\mathbf{f}_{\mathbf{z}_i}||_p \leq 1\ i \in \mathbb{N}_1^{L_\mathbf{f}}, \tag{21a}$$

$$\|\mathbf{A}^\mathbf{g}_j\|_p < 1,\ ||\mathbf{\Lambda}^\mathbf{g}_{\mathbf{z}_j}||_p \leq 1,\ j \in \mathbb{N}_1^{L_\mathbf{g}}, \tag{21b}$$

$$\forall \mathbf{x} \in \textit{Domain}(\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x}), \mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})). \tag{21c}$$

*Proof.* First we show the sufficiency $\|\mathbf{A}_i\|_p < 1$ of contractive weights and non-expanding activation scaling matrices $\|\mathbf{\Lambda}_{\mathbf{z}_i}\| \leq 1$ to guarantee the contractivity of arbitrary deep neural networks. Assuming general non-square weights $\mathbf{A}_i \in \mathbf{R}^{n_i \times m_i}$ we use the submultiplicativity of the induced $p$-norms to upper bound the norm of a products of $m$ matrices given as:

$$\|\mathbf{A}_1 \dots \mathbf{A}_m\|_p \leq \|\mathbf{A}_1\|_p \dots \|\mathbf{A}_m\|_p \tag{22}$$

Now by applying (22) to the linear parts (3) of the mean neural network $\mathbf{f}_{\theta_\mathbf{f}}$ in the pointwise affine form (2) with $\|\mathbf{A}^\mathbf{f}_i\|_p < 1,\ \forall i \in \mathbb{N}_0^L$, $\|\mathbf{\Lambda}^\mathbf{f}_{\mathbf{z}_j}\|_p \leq 1,\ \forall j \in \mathbb{N}_1^L$, it yields $\|\mathbf{A}_\mathbf{f}(\mathbf{x})\|_p < 1$ over the entire domain of $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$, thus with $p = 2$ satisfying the contraction condition $||\mathbf{A}_\mathbf{f}(\mathbf{x})||_2 < 1$ for affine maps. The submultiplicativity (22) naturally applies also to the variance network $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ thus implying the contraction towards a fixed point given the conditions (21). □

**Remark 6.** *To guarantee stochastic stability, we can relax the upper bound of the second moment as* $K = \prod_i^L c^\mathbf{A} c^\mathbf{\Lambda}$*, where* $c^\mathbf{A} > 0$*, and* $c^\mathbf{\Lambda} > 0$ *represent the relaxed upper bounds of the operator norms in condition* (21b)*. Thus satisfying the relaxed boundedness condition on the variance via* (15)*.*

Assuming the contraction conditions (17) or (21) hold and the mean neural network $\mathbf{f}_{\theta_\mathbf{f}}$ has zero bias, then the DMM's mean network $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ is equivalent with stable parameter varying linear map (3) with equilibrium in the origin, i.e. $\bar{\mathbf{x}} = \mathbf{0}$. In the case with non-zero bias in $\mathbf{f}_{\theta_\mathbf{f}}$, the corresponding PWA map (2) has non-zero equilibrium $\bar{\mathbf{x}} \neq \mathbf{0}$. Both conditions (17) or (21) are sufficient for a convergence of a DMM (12) to a stable equilibrium $\bar{\mathbf{x}}$. However, they do not provide bounds of the admissible values of the equilibrium $\bar{\mathbf{x}}$. The corresponding equilibrium bounds are provided in the supplementary material.

### 3.4 Design of Stable Deep Markov Models

In this section, we provide a set of practical design methods for provably stable DMM (12). Based on the Corollary 5 the use of contractive activation functions together with contractive weights for both mean and variance network will guarantee the stability of DMM by design. In particular, the conditions (21) on bounded norm of transitions' activation scaling matrices $||\mathbf{\Lambda}^\mathbf{f}_{\mathbf{z}_i}|| \leq 1$, $||\mathbf{\Lambda}^\mathbf{g}_{\mathbf{z}_j}|| \leq 1$ implies Lipschitz continuous activation functions with constant $\mathcal{K} \leq 1$. Conveniently, this condition is satisfied for many popular activation functions such as `ReLU`, `LeakyReLU`, or `tanh`. The contractivity conditions (21) on weight matrices $\|\mathbf{A}^\mathbf{f}_i\|_p < 1$, $\|\mathbf{A}^\mathbf{g}_j\|_p < 1$, respectively, can be enforced by employing various matrix factorizations proposed in the deep neural network litertature. Examples include singular value decomposition (SVD) [Zhang et al., 2018], Perron-Frobenius (PF) [Tuor et al., 2020], and Gershgorin discs (GD) [Lechner et al., 2020] factorizations given below.

**PF weights:** This factorization applies Perron-Frobenius theorem for constraining the dominant eigenvalue of the square nonnegative matrices. Based on this theorem, we can construct the weight matrix $\mathbf{A}$ with bounded eigenvalues as follows:

$$\mathbf{M} = \lambda_{\text{max}} - (\lambda_{\text{max}} - \lambda_{\text{min}})g(\mathbf{M}') \tag{23a}$$

$$\mathbf{A}_{i,j} = \frac{\exp(\mathbf{A'}_{ij})}{\sum_{k=1}^{n_x} \exp(\mathbf{A'}_{ik})}\mathbf{M}_{i,j} \tag{23b}$$

here $\mathbf{M}$ represents the damping factor parameterized by the matrix $\mathbf{M}' \in \mathbb{R}^{n_x \times n_x}$, while $\mathbf{A}' \in \mathbb{R}^{n_x \times n_x}$ represents the second parameter matrix encoding the stable weights $\mathbf{A}$. The lower and upper bound of the dominant eigenvalue are given by $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$, respectively.

**SVD weights:** Inspired by singular value decomposition (SVD), this method decomposes a possibly non-square weight matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ into two unitary matrices $\mathbf{U}$ and $\mathbf{V}$, and a diagonal matrix $\mathbf{\Sigma}$ with singular values on its diagonal. The orthogonality of $\mathbf{U}$ and $\mathbf{V}$ is enforced via penalties:

$$\mathcal{L}_{\text{reg}} = ||\mathbf{I} - \mathbf{U}\mathbf{U}^\top||_2 + ||\mathbf{I} - \mathbf{U}^\top\mathbf{U}||_2 + ||\mathbf{I} - \mathbf{V}\mathbf{V}^\top||_2 + ||\mathbf{I} - \mathbf{V}^\top\mathbf{V}||_2 \tag{24}$$

An alternative approach to penalties introduced in Zhang et al. [2018] is to use Householder reflectors to represent unitary matrices $\mathbf{U}$ and $\mathbf{V}$. The constraints $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$ on the singular values $\lambda$ can be implemented by clamping and scaling given as:

$$\mathbf{\Sigma} = \text{diag}(\lambda_{\text{max}} - (\lambda_{\text{max}} - \lambda_{\text{min}}) \cdot \sigma(\lambda)) \tag{25}$$

**GD weights:** This method supporting square matrices leverages the Gershgorin dics theorem [Varga, 2004]. It says that all eigenvalues $\lambda_i$ of the weight $\mathbf{A}$ can be bounded in the complex plane with center $\lambda$ and radius $r$ given by the formula:

$$\mathbf{A} = \texttt{diag}\left(\tfrac{r}{s_1}, ..., \tfrac{r}{s_n}\right)\mathbf{M} + \texttt{diag}\left(\lambda, ..., \lambda\right) \tag{26}$$

Where $\mathbf{M} \in \mathbb{R}^{n \times n}$ with $m_{i,j} \sim \mathcal{U}(0, 1)$, except $m_{i,i} = 0$ is learnable parameter matrix. While diagonal matrices $\texttt{diag}\left(\tfrac{r}{s_1}, ..., \tfrac{r}{s_n}\right)$, and $\texttt{diag}\left(\lambda, ..., \lambda\right)$ represent radii and centers of the bounded eigenvalues, where $s_j = \sum_{i \neq j} m_{i,j}$.

**Parametric stability constraints:** The disadvantage of enforcing the global stability conditions as given via Corollary 5 is their negative effect on the expressivity of the DMM, resulting in dynamics with a single point or line attractors. This will effectively prevent the DMM from expressing more complex attractors such limit cycles or chaotic attractors. As a more expressive alternative we introduce the use of parameter varying bounds in the conditions (17), and (21), such as:

$$\underline{\mathbf{p}}(\mathbf{x}) < \|\mathbf{A}_\mathbf{f}(\mathbf{x})\|_p < \overline{\mathbf{p}}(\mathbf{x}) \tag{27}$$

Where $\underline{\mathbf{p}}(\mathbf{x}) : \mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}$, and $\overline{\mathbf{p}}(\mathbf{x}) : \mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}$ are scalar valued functions parametrizing lower and upper bounds of operator norm of the DMM's mean transition dynamics. Similar parametric constraints can be applied to the variance bounds in (17), or weight norm constraints in (21). This approach allows us to control the contractivity of the DMMs depending on the position in the state space. This allows us to increase the expressivity of the DMM, e.g., by partitioning the state space into constrained and unconstrained regions resulting in DMM with hybrid or switching dynamics. In particular, we could divide the state space to outer contractive regions (where conditions (17) hold) and inner relaxed regions allowing for more complex trajectories to emerge. This parametrization will effectively generate non-empty attractor set in which it is possible to learn arbitrary attractor shape. The proposed state space partitioning method is inspired by the Bendinxon-Dulac criteria on periodic solutions of differential equations [McCluskey and Muldowney, 1998].

## 4 Numerical Case Studies

In this section we empirically validate the conditions given in Theorem 3 and Corollary 5 by investigating the dynamics of DMM's transition maps (12) whose mean $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ and variance $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ are parametrized by neural networks with different spectral distributions of their weights and activation scaling matrices (6). We apply spectral analysis to the PWA forms (2) of neural networks modeling the mean and variance maps to obtain the corresponding spectra of DMMs. We performed the experiments using the probabilistic programming language Pyro [Bingham et al., 2019].

### 4.1 Design of the Experiments

Since the stability of DMM (12) depends on the transition dynamcis, in all of the case studies we consider a fully observable model with identity as an emission map. We parametrize the mean and variances of the transition map $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ (13) and $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ (14) by feedforward neural networks. Given the mean neural network $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ we generate a set of different transition dynamcis by changing activation functions $\mathbf{v}(x) \in \{\texttt{ReLU}, \texttt{Tanh}, \texttt{Sigmoid}, \texttt{SELU}, \texttt{Softplus}\}$, layer depth $L \in \{1, 2, 4, 8\}$, and presence of bias $b \in \{\texttt{True}, \texttt{False}\}$. For the variance network $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ we use $\texttt{ReLU}$ activations. For both, mean and variance networks we initialize their weights with desired spectral properties via design methods described in Section 3.4. In particular we use SVD, PF, and GD factorizations to bound the weight's singular values in a prescribed range. We generate three categories of weights $\mathbf{A}_i$: stable with operator norm strictly below one $||\mathbf{A}_i||_p < 1$, marginally stable with norm close one $||\mathbf{A}_i||_p \approx 1$, and unstable with norm larger than one $||\mathbf{A}_i||_p > 1$.

### 4.2 Stability Analysis of Deep Markov Models

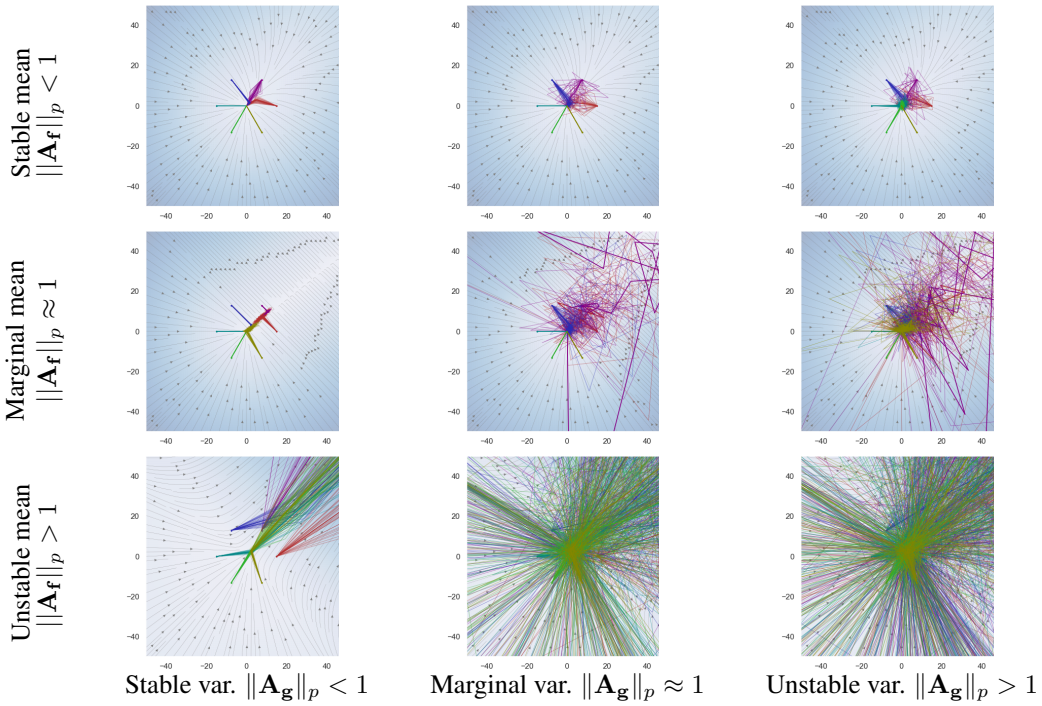


*Figure 1: Phase portraits of DMMs demonstrating the effect of norm bounds on mean $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ and variance $\mathbf{f}_{\theta_\mathbf{g}}(\mathbf{x})$ networks modeling transition dynamics. Thin lines are samples of the stochastic dynamics with bold lines representing mean trajectories. Colors represent different initial conditions.*

In order to provide intuitive visualisations of the dynamics in the phase space, in this section, we focus on two dimensional system. Fig. 1 visualizes the phase portraits of randomly generated DMM's probabilistic transition maps of the mean $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ and variance $\mathbf{f}_{\theta_\mathbf{g}}(\mathbf{x})$ networks with constrained operator norms enforced using PF weights from Section 3.4. Figures in the first row demonstrate that DMMs with asymptotically stable mean transition dynamics $||\mathbf{A_f}||_p < 1$ with bounded variances $\|\mathbf{A_g}\|_p < K,\ K > 0$ generate stable single point attractors. Hence they validate the sufficient conditions of Theorem 3. Figures in the second row display dynamics of DMM with marginally stable mean $||\mathbf{A_f}||_p \approx 1$. Due to the non-dissipativeness of the mean transition dynamics, the trajectories converge to a line attractor only if the variance is a converging map $\|\mathbf{A_g}\|_p < 1$, thus having a dissipative second moment. In case of marginally stable variance, $\|\mathbf{A_g}\|_p \approx 1$ the energy conserving nature of the mean and variance together generate random walk type trajectories along the direction of the mean's line attractor. For the cases with unstable variance $\|\mathbf{A_g}\|_p > 1$, the overall dynamics behaves close to a Brownian motion with a degree of randomness, which is positively

correlated with the variance network's operator norm. Figures in the third row show diverging dynamics of DMM with unstable mean $||\mathbf{A_f}||_p > 1$. With converging variance $\|\mathbf{A_g}\|_p < 1$ the diverging stochastic trajectories stay close to the mean direction. While, for both marginal and unstable variances the stochastic trajectories diverge in all directions.

### 4.3 Effect of Biases and Depth on the Stability of Deep Markov Models

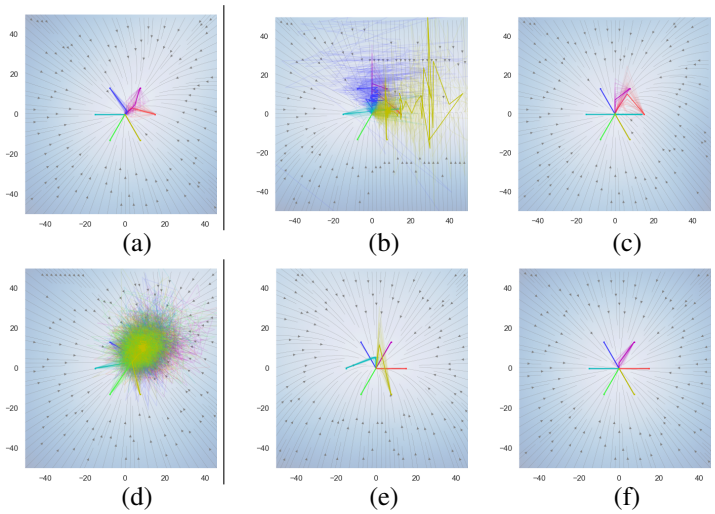


*Figure 2: Left panels show the effect of biases using PF regularization and* `ReLU` *activation ((a) w/o bias, (d) w bias). Right panels show the effect of network* $\mathbf{f}$ *depths with SVD regularization and* `ReLU` *: (b)* $1$ *layer, (c)* $2$ *layers, (e)* $4$ *layers, (f)* $8$ *layers.*

In Fig. 2, we experiment with biases and depths of mean $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x})$ and variance $\mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ networks.

**Effect of biases:** In the left panels of Fig. 2, we demonstrate the dynamics of DMM with `tanh` activations and SVD factorized weights resulting in asymptotically stable transition maps, thus $||\mathbf{A_f}(\mathbf{x})|| < 1$, and $||\mathbf{A_g}(\mathbf{x})|| < 1$. Fig. 2(a) shows the scenario without any bias whereas Fig. 2(d) shows the scenario where both $\mathbf{f}_{\theta_\mathbf{f}}(\mathbf{x}), \mathbf{g}_{\theta_\mathbf{g}}(\mathbf{x})$ have bias terms. It demonstrates that the general contractive nature of the stable behavior, as given via conditions (17), does not change with addition of biases. Instead, the biases shift the region of attraction by generating non-zero equilibrium points. This shift is correlated with absolute value of the aggregate bias term of the PWA form (2). For the norm bounds on the equilibria of stable DMM see supplementary material.

**Effect of depth:** The right panels of Fig. 2 demonstrate the dynamics of DMMs with increasing number of layers using `ReLU` activations and SVD weights close to marginal stability $0.99 < ||\mathbf{A}_i(\mathbf{x})|| < 1$. It can be seen that with increase in the number of layers, the convergence of trajectories toward origin becomes less uncertain. In this case, a larger number of mildly contractive layers results in stabilizing behavior, demonstrating the effect of the norm submultiplicativity on the operator norm of the mean transition dynamics (22). However, one needs to be careful about the delicate balance between stability, and ability to efficiently train the parameters of DMM with gradient-based optimization. With increasing depth, the norm product of the contractive layers (22) will eventually result in a network with a very small operator norm thus causing the vanishing gradient problem. Analogously, the exploding gradient problem will occur for DMM parametrized with very deep neural networks with non-contractive layers, i.e. $||\mathbf{A}_i(\mathbf{x})|| > 1$ The use of parametric stability constraints (27) as a function of depth could be an efficient strategy for avoiding the vanishing and exploding gradients by keeping the overall dynamics norm bounded.

### 4.4 Deep Markov Models with Parametrized Stability Constraints

In Fig. 3 we demonstrate the use of parametrized stability constraints (27) in the design of stable DMMs without compromising the expressivity as it is in the case of restrictive single point attractors

enforced via (17) and (21). In particular, we design two DMMs with randomly generated weights with three phase space regions with different mean transition dynamics, (i) an inner expanding region $||\mathbf{A_f}(\mathbf{x})|| > 1,\ \mathbf{x}^i \in \mathcal{R}_1$, (ii) a middle marginal region $||\mathbf{A_f}(\mathbf{x}^i)|| \approx 1,\ \mathbf{x}^i \in \mathcal{R}_2$, and (iii) an outer contractive region $||\mathbf{A_f}(\mathbf{x}^i)|| < 1,\ \mathbf{x}^i \in \mathcal{R}_3$. Where the regions are given as $\mathcal{R}_1 = \{\mathbf{x}|0 \leq ||\mathbf{x}||_2 < 20\}$, $\mathcal{R}_2 = \{\mathbf{x}|20 \leq ||\mathbf{x}||_2 < 40\}$, and $\mathcal{R}_3 = \{\mathbf{x}|40 \leq ||\mathbf{x}||_2\}$, respectively. The variance dynamics in both cases is kept being contractive $||\mathbf{A_g}(\mathbf{x}^i)|| < 1$. From Fig. 3 it is apparent that the overall dynamics of the DMMs with parametrized constraints (27) is able to generate stochastic periodic behavior while remaining bounded within prescribed region of attraction, thus providing high degree of expressivity while being provable stable. On the left (Fig. 3 (a) and (c)) we show phase plots, and on the right (Fig. 3 (b) and (d)) corresponding time series trajectories. As a potential extension, we envision learning the constraints bounds $\underline{\mathbf{p}}(\mathbf{x})$, and $\overline{\mathbf{p}}(\mathbf{x})$ in (27) using penalty methods.

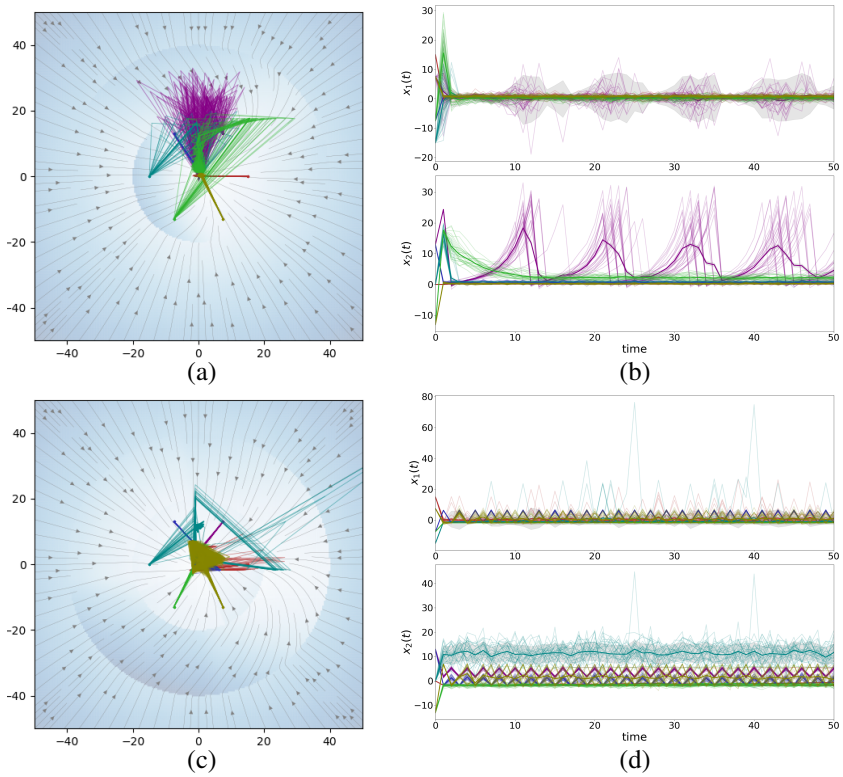


*Figure 3: Phase plots (left panels), and time series trajectories (right panels) of DMMs with parametrized stability constraints exhibiting periodic behavior within bounded regions of attaction. Both cases consider SVD regularizations with* `softplus` *(top) and* `SELU`*(bottom) respectively.*

## 5 Conclusion

In this paper, we introduce a new stability analysis method for deep Markov models (DMMs). As the main result, we provide sufficient conditions for the stochastic stability and introduce a set of practical methods for designing provably stable DMMs. In particular, we discuss the use of contractive weight matrices factorizations and stability conditions for the activation functions. Furthermore, we propose using novel parametric stability constraints allowing expression of more complex stochastic dynamics while remaining contractive towards non-empty region of attraction. The proposed theory is supported by numerical experiments, with design guidelines considering weight factorizations, choices of activation functions, network depth, or use of the bias terms for guaranteed stability. In future work, we aim to derive the stability guarantees for a broader family of probability distributions modeled by normalizing flows. We also aim to expand the theory to partially observable Markov decision processed (POMDP) to derive closed-loop stability guarantees in the context of deep reinforcement learning.

## Acknowledgments and Disclosure of Funding

We acknowledge our colleagues Aaron Tuor, Mia Skomski, Soumya Vasisht, and Draguna Vrabie for their contributions to related topics that served as a base for developing the method presented in this paper. We would like to thank Craig Baker and David Rolnick for fruitful discussions that helped improve the technical quality of the presented ideas. Also, we want to thank our anonymous reviewers for their constructive feedback and suggestions.

This research was supported by the U.S. Department of Energy, through the Office of Advanced Scientific Computing Research's "Data-Driven Decision Control for Complex Systems (DnC2S)" project. Pacific Northwest National Laboratory is operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract No. DE-AC05-76RL01830. Oak Ridge National Laboratory is operated by UT-Battelle LLC for the U.S. Department of Energy under contract number DE-AC05-00OR22725.

## 6 Limitations and Broader Impact

Stability is the major concern of many safety-critical systems. The significance of the presented work lies in the proposed stability conditions, analysis, and design methods allowing the generation of provably stable DMMs. Furthermore, the presented methods for the design of stable DMMs could be of large significance in the context of stochastic control systems with stability guarantees. Thus, the authors believe that besides academic relevance, the methods presented in this paper have the potential for practical impact in many real-world applications such as unmanned autonomous vehicles, robotics, or process control applications.

The authors are aware of the limitations of the presented numerical case studies focusing on small-scale DMMs with two-dimensional state space. This choice was made for the sake of the visualizations of the state space trajectories allowing us to provide intuitive examples of the presented theoretical results. As part of the future work, the authors plan to use the proposed stability analysis and design methods for DMMs on real-world datasets with higher dimensional state space.

The presented work falls into the basic research category. As such, authors are not aware of any potential direct negative societal impact of the proposed work. On the contrary, authors of this paper believe that the presented theory is a minor contribution towards general knowledge, which accumulation has been historically proved to inherently benefit all humanity.

## References

M. Awiszus and B. Rosenhahn. Markov chain neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2180–2187, 2018.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. *Advances in neural information processing systems*, 1:577–584, 2002.

P. Becker, H. Pandya, G. H. W. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. *CoRR*, abs/1905.07357, 2019. URL `http://arxiv.org/abs/1905.07357`.

T. Beckers and S. Hirche. Stability of gaussian process state space models. In *2016 European Control Conference (ECC)*, pages 2275–2281. IEEE, 2016.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

R. Brayton and C. Tong. Stability of dynamical systems: A constructive approach. *IEEE Transactions on Circuits and Systems*, 26(4):224–234, 1979. doi:10.1109/TCS.1979.1084637.

O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer Science & Business Media, 2006.

Z. Che, S. Purushotham, G. Li, B. Jiang, and Y. Liu. Hierarchical deep generative models for multi-rate multivariate time series. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 784–793, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/che18a.html.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 6571–6583. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

M. Ciccone, M. Gallieri, J. Masci, C. Osendorfer, and F. J. Gomez. Nais-net: Stable deep networks from non-autonomous differential equations. *CoRR*, abs/1804.07209, 2018. URL http://arxiv.org/abs/1804.07209.

M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

S. R. Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

S. Eleftheriadis, T. Nicholson, M. P. Deisenroth, and J. Hensman. Identification of gaussian process state space models. In *NIPS*, pages 5309–5319, 2017.

N. Elia, J. Wang, and X. Ma. Mean square limitations of spatially invariant networked systems. In *Control of Cyber-Physical Systems*, pages 357–378. Springer, 2013.

R. Engelken, F. Wolf, and L. Abbott. Lyapunov spectra of chaotic recurrent neural networks. *arXiv preprint arXiv:2006.02427*, 2020.

R. E. Farmer, D. F. Waggoner, and T. Zha. Understanding markov-switching rational expectations models. *Journal of Economic theory*, 144(5):1849–1867, 2009.

T. A. Ferreira. Reinforced deep markov models with applications in automatic trading. *arXiv preprint arXiv:2011.04391*, 2020.

M. Fraccaro, S. K. Sønderby, U. Paquet, and O. Winther. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine learning*, 29(2):245–273, 1997.

S. Goel and A. Klivans. Eigenvalue decay implies polynomial-time learnability for neural networks. In *Advances in Neural Information Processing Systems*, pages 2192–2202, 2017.

S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. *CoRR*, abs/1906.01563, 2019. URL http://arxiv.org/abs/1906.01563.

E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1): 014004, 2017. URL http://arxiv.org/abs/1705.03341.

E. Haber, K. Lensink, E. Treister, and L. Ruthotto. Imexnet: A forward stable deep neural network. *CoRR*, abs/1903.02639, 2019. URL http://arxiv.org/abs/1903.02639.

T. Huster, C.-Y. J. Chiang, and R. Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 16–29. Springer, 2018.

V. John, A. Boyali, and S. Mita. Gabor filter and gershgorin disk-based convolutional filter constraining for image classification. *Int. J. Mach. Learn. Comput*, 7(4):55–60, 2017.

H. Khalil. *Nonlinear Systems*. Prentice-Hall, New York, 2002.

R. Khasminskii. *Stochastic Stability of Differential Equations*, volume 66. Springer, 2012. doi:10.1007/978-3-642-23280-0.

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

R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

M. Lechner, R. Hasani, D. Rus, and R. Grosu. Gershgorin loss stabilizes the recurrent neural network compartment of an end-to-end robot learning scheme. In *2020 International Conference on Robotics and Automation (ICRA). IEEE*, 2020.

L. Li, Y. Zhao, D. Jiang, Y. Zhang, F. Wang, I. Gonzalez, E. Valentin, and H. Sahli. Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 312–317, 2013. doi:10.1109/ACII.2013.58.

D. Liu, A. Honoré, S. Chatterjee, and L. K. Rasmussen. Powering hidden markov model by neural network based generative models. *arXiv preprint arXiv:1910.05744*, 2019.

J. Lu and R. E. Skelton. Mean-square small gain theorem for stochastic control: discrete-time case. *IEEE Transactions on Automatic Control*, 47(3):490–494, 2002.

O. Ludwig, U. Nunes, and R. Araujo. Eigenvalue decay: A new method for neural network regularization. *Neurocomputing*, 124:33–42, 2014.

G. Manek and J. Z. Kolter. Learning stable deep dynamics models. In *Advances in Neural Information Processing Systems 32*, pages 11126–11134. 2019. URL http://papers.nips.cc/paper/9292-learning-stable-deep-dynamics-models.pdf.

C. C. McCluskey and J. S. Muldowney. Bendixson-dulac criteria for difference equations. 10(4): 567–575, 1998. ISSN 1572-9222. doi:10.1023/A:1022677008393. URL https://doi.org/10.1023/A:1022677008393.

P. McLane. Optimal stochastic control of linear systems with state-and control-dependent disturbances. *IEEE Transactions on Automatic Control*, 16(6):793–798, 1971.

Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *International Conference on Machine Learning*, pages 2401–2409. PMLR, 2017.

G. Montanez, S. Amizadeh, and N. Laptev. Inertial hidden markov models: Modeling change in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

M. K. Mustafa, T. Allen, and K. Appiah. A comparative review of dynamic neural networks and hidden markov model methods for mobile on-device speech recognition. *Neural Computing and Applications*, 31(2):891–899, 2019.

S. P. Nandanoori, A. Diwadkar, and U. Vaidya. Mean square stability analysis of stochastic continuous-time linear networked systems. *IEEE Transactions on Automatic Control*, 63(12):4323–4330, 2018.

B. H. Prasetio, H. Tamura, and K. Tanno. Deep time-delay markov network for prediction and modeling the stress and emotions state transition. 10(1):18071. ISSN 2045-2322. doi:10.1038/s41598-020-75155-w. URL https://doi.org/10.1038/s41598-020-75155-w.

M. Qu, Y. Bengio, and J. Tang. Gmnn: Graph markov neural networks. In *International conference on machine learning*, pages 5241–5250. PMLR, 2019.

L. Rabiner and B. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. *CoRR*, abs/1711.10561, 2017. URL http://arxiv.org/abs/1711.10561.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

S. D. Shashua and S. Mannor. Deep robust kalman filter. *CoRR*, abs/1703.02310, 2017. URL http://arxiv.org/abs/1703.02310.

Z. Tan, H. Soh, and D. C. Ong. Factorized inference in deep markov models for incomplete multimodal time series. *CoRR*, abs/1905.13570, 2019. URL http://arxiv.org/abs/1905.13570.

S. Toyer, A. Cherian, T. Han, and S. Gould. Human pose forecasting via deep markov models, 2017.

K. M. Tran, Y. Bisk, A. Vaswani, D. Marcu, and K. Knight. Unsupervised neural hidden Markov models. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 63–71, Austin, TX, Nov. 2016. Association for Computational Linguistics. doi:10.18653/v1/W16-5907. URL https://www.aclweb.org/anthology/W16-5907.

A. Tuor, J. Drgona, and D. Vrabie. Constrained neural ordinary differential equations with stability guarantees. *arXiv preprint arXiv:2004.10883*, 2020.

R. Varga. *Geršgorin and His Circles*, volume 36. Springer, Berlin, Heidelberg, 01 2004. doi:10.1007/978-3-642-17798-9.

R. Vogt, M. P. Touzel, E. Shlizerman, and G. Lajoie. On lyapunov exponents for rnns: Understanding information propagation using dynamical systems tools. *arXiv preprint arXiv:2006.14123*, 2020.

S. Wang, J. Xiang, Y. Zhong, and Y. Zhou. Convolutional neural network-based hidden markov models for rolling element bearing fault identification. *Knowledge-Based Systems*, 144:65 – 76, 2018. ISSN 0950-7051. doi:https://doi.org/10.1016/j.knosys.2017.12.027. URL http://www.sciencedirect.com/science/article/pii/S0950705117306056.

J. L. Willems and J. C. Willems. Feedback stabilizability for stochastic systems with state and control dependent noise. *Automatica*, 12(3):277 – 283, 1976. ISSN 0005-1098. doi:https://doi.org/10.1016/0005-1098(76)90029-7. URL http://www.sciencedirect.com/science/article/pii/0005109876900297.

B. Wu, M. Cubuktepe, and U. Topcu. Switched linear systems meet markov decision processes: Stability guaranteed policy synthesis. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2509–2516. IEEE, 2019.

H. Wu, A. Mardt, L. Pasquali, and F. Noe. Deep generative markov state models. *arXiv preprint arXiv:1805.07601*, 2018.

J. Zhang, Q. Lei, and I. Dhillon. Stabilizing gradients for deep neural networks via efficient svd parameterization. In *International Conference on Machine Learning*, pages 5806–5814, 2018.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We believe the abstract and the introduction are accurate representation of the paper and that the claims are supported by the presented methods and case studies.
   (b) Did you describe the limitations of your work? [Yes] Please see the limitations section.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We presents fundamental theoretical work, which by itself has no negative implications towards societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Please see the definitions in the methodology section and in the supplementary material.

(b) Did you include complete proofs of all theoretical results? [Yes] Please see the methodology section and supplementary material.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The authors plan to open-source the code with the camera ready version of the paper. The reason why we do not disclose the case study code during the review process is the use of custom open-source libraries that could reveal the affiliation of the authors.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide the summary of the experiment hyperparameters in the supplementary material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We performed the experiments using the probabilistic programming language Pyro Bingham et al. [2019] which we cite in the paper.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]