Dynamical Variational Autoencoders : discrete-time and continuous-time models. Links to stochastic calculus and stochastic differential equations

Benjamin Deporte : benjamin.deporte@ens-paris-saclay.fr

August 2025 - DRAFT

Summary

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t \in \mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t \in \mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t \in \mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t\in\mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t \in \mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t\in\mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Abstract

- **Data sequences** : we consider data sequences $(X_t)_{t \in \mathbb{T}} \in \mathbb{R}^D$, where $\mathbb{T}$ is a set of times, either discrete or continuous. ie : time-series, videos, motion captures, patient data...
- **Dynamical Variational Auto Encoders** are a class of VAE models in which some structure is given to the latent variables to express the time dependency of the X_t.
- **Discrete-time DVAEs** are a large set of models, from the well-known Kalman filter up to the Variational RNN. We review the Deep Kalman filter and the VRNN models.
- **Continuous-time DVAEs** use a continuous prior over the latent variables, which allows to deal with irregularly sampled data, or data with missing components. We review the Gaussian Process VAE.
- **Stochastic calculus and stochastic differential equations** provides an elegant mathematical framework for DVAEs. We give a survival kit on stochastic calculus and SDEs.
- **The solution of a linear SDE is a Gaussian process**. We can use known filtering and smoothing Kalman algorithms to compute the GP regression (ie posterior distribution) in GP-VAE with a linear cost.
- **Beyond GP : Latent SDE model** - Not all Gaussian Processes are the solution to a linear SDE. Also, if the solution of a general SDE is a Markov process, it is not necessarily a Gaussian process. This leads to considering Latent SDE model, where the latent prior is a general SDE.

Dynamical Variational Auto Encoders : what is it ?

- **Dynamical Variational Auto Encoders** are a class of VAEs in which some structure is given to the latent variables to encode the time dependency.
- DVAEs can be discrete-time or continuous models, can require regularly-sampled data, or can manage irregularly sampled data.
- For example, a Kalman filter is the simplest DVAE :
  - first order Markov chain for latent variables
  - linear Gaussian observation model.
- As in vanilla VAEs, inference is performed by evidence lower bound maximization.
- Notations

  - the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,\dots,T}\} \in \mathbb{R}^F$.
  - the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,\dots,T}\} \in \mathbb{R}^L$
  - optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,\dots,T}\} \in \mathbb{R}^U$

Dynamical Variational Auto Encoders : what is it ?

- **Dynamical Variational Auto Encoders** are a class of VAEs in which some structure is given to the latent variables to encode the time dependency.
- DVAEs can be discrete-time or continuous models, can require regularly-sampled data, or can manage irregularly sampled data.
- For example, a Kalman filter is the simplest DVAE :
  - first order Markov chain for latent variables
  - linear Gaussian observation model.
- As in vanilla VAEs, inference is performed by evidence lower bound maximization.
- Notations

  - the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,\dots,T}\} \in \mathbb{R}^F$.
  - the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,\dots,T}\} \in \mathbb{R}^L$
  - optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,\dots,T}\} \in \mathbb{R}^U$

Dynamical Variational Auto Encoders : what is it ?

- **Dynamical Variational Auto Encoders** are a class of VAEs in which some structure is given to the latent variables to encode the time dependency.
- DVAEs can be discrete-time or continuous models, can require regularly-sampled data, or can manage irregularly sampled data.
- For example, a Kalman filter is the simplest DVAE :
  - first order Markov chain for latent variables
  - linear Gaussian observation model.
- As in vanilla VAEs, inference is performed by evidence lower bound maximization.
- Notations

  - the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,\dots,T}\} \in \mathbb{R}^F$.
  - the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,\dots,T}\} \in \mathbb{R}^L$
  - optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,\dots,T}\} \in \mathbb{R}^U$

Dynamical Variational Auto Encoders : what is it ?

- **Dynamical Variational Auto Encoders** are a class of VAEs in which some structure is given to the latent variables to encode the time dependency.
- DVAEs can be discrete-time or continuous models, can require regularly-sampled data, or can manage irregularly sampled data.
- For example, a Kalman filter is the simplest DVAE :
  - first order Markov chain for latent variables
  - linear Gaussian observation model.
- As in vanilla VAEs, inference is performed by evidence lower bound maximization.
- Notations

  - the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,\dots,T}\} \in \mathbb{R}^F$.
  - the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,\dots,T}\} \in \mathbb{R}^L$
  - optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,\dots,T}\} \in \mathbb{R}^U$

Dynamical Variational Auto Encoders : what is it ?

- **Dynamical Variational Auto Encoders** are a class of VAEs in which some structure is given to the latent variables to encode the time dependency.
- DVAEs can be discrete-time or continuous models, can require regularly-sampled data, or can manage irregularly sampled data.
- For example, a Kalman filter is the simplest DVAE :
  - first order Markov chain for latent variables
  - linear Gaussian observation model.
- As in vanilla VAEs, inference is performed by evidence lower bound maximization.
- Notations

> - the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,\ldots,T}\} \in \mathbb{R}^F$.
> - the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,\ldots,T}\} \in \mathbb{R}^L$
> - optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,\ldots,T}\} \in \mathbb{R}^U$

General formulation of DVAE

Generative model

$$
\begin{aligned}
p(x_{1:T}, z_{1:T} | u_{1:T}) &= \prod_{t=1}^{T} p(x_t, z_t | x_{1:t-1}, z_{1:t-1}, u_{1:T}) \\
&= \prod_{t=1}^{T} p(x_t | x_{1:t-1}, z_{1:t}, u_{1:T}) p(z_t | x_{1:t-1}, z_{1:t-1}, u_{1:T}) \\
&= \prod_{t=1}^{T} p(x_t | x_{1:t-1}, z_{1:t}, u_{1:t}) p(z_t | x_{1:t-1}, z_{1:t-1}, u_{1:t})
\end{aligned}
$$

The only assumption that is made is a a causal dependency of the x_t, z_t on the inputs $u_{1:t}$, thus allowing to change the conditioning $|u_{1:T}$ into $|u_{1:t}$.
In the rest of the presentation, we will consider systems with no input, and drop the conditioning on $u_{1:t}$ to simplify notations. However, the reasoning remains the same with inputs.

Posteriors

- The true posterior $p(z_{1:T}|x_{1:T})$ is usually untractable, but can be developed:

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

- As in vanilla Variational Auto Encoders (VAEs), the inference model is the approximation of the true posterior by an parametric encoder $q_\phi(z_{1:T}|x_{1:T})$, where ϕ is the set of parameters:

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})$$

- Depending on the chosen graphical models and the corresponding D-separation results, the observation model $p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$ (with θ_x the set of parameters of the observation model) and approximate posterior $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ will simplify.
- It is also considered a good practice to copy the expression of $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ from the expression of the true posterior resulting from the D-separation analysis (see next chapters for examples).

Posteriors

- The true posterior $p(z_{1:T}|x_{1:T})$ is usually untractable, but can be developed:

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

- As in vanilla VAEs, the inference model is the approximation of the true posterior by an parametric encoder $q_\phi(z_{1:T}|x_{1:T})$, where ϕ is the set of parameters:

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})$$

- Depending on the chosen graphical models and the corresponding D-separation results, the observation model $p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$ (with θ_x the set of parameters of the observation model) and approximate posterior $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ will simplify.
- It is also considered a good practice to copy the expression of $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ from the expression of the true posterior resulting from the D-separation analysis (see next chapters for examples).

Posteriors

- The true posterior $p(z_{1:T}|x_{1:T})$ is usually untractable, but can be developed:

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

- As in vanilla VAEs, the inference model is the approximation of the true posterior by an parametric encoder $q_\phi(z_{1:T}|x_{1:T})$, where ϕ is the set of parameters:

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})$$

- Depending on the chosen graphical models and the corresponding D-separation results, the observation model $p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$ (with θ_x the set of parameters of the observation model) and approximate posterior $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ will simplify.
- It is also considered a good practice to copy the expression of $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ from the expression of the true posterior resulting from the D-separation analysis (see next chapters for examples).

Posteriors

- The true posterior $p(z_{1:T}|x_{1:T})$ is usually untractable, but can be developed:

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

- As in vanilla VAEs, the inference model is the approximation of the true posterior by an parametric encoder $q_\phi(z_{1:T}|x_{1:T})$, where ϕ is the set of parameters:

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})$$

- Depending on the chosen graphical models and the corresponding D-separation results, the observation model $p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$ (with θ_x the set of parameters of the observation model) and approximate posterior $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ will simplify.
- It is also considered a good practice to copy the expression of $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ from the expression of the true posterior resulting from the D-separation analysis (see next chapters for examples).

Likelihood

- Observation model and encoder:

$$p_\theta(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t | x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t | z_{1:t-1}, x_{1:t-1}) \tag{1}$$

$$q_\phi(z_{1:T} | x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t | z_{1:t-1}, x_{1:T}) \tag{2}$$

- Log likelihood

$$\log p(x_{1:T}) = \log \frac{p(x_{1:T}, z_{1:T})}{p(z_{1:T}|x_{1:T})} \tag{3}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} \frac{q_\phi(z_{1:T}|x_{1:T})}{p(z_{1:T}|x_{1:T})} \tag{4}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} + \mathbb{KL}\left(q_\phi(z_{1:T}|x_{1:T}) || p(z_{1:T}|x_{1:T})\right) \tag{5}$$

$$\geq \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} = \mathcal{L}(\theta, \phi, X) \tag{6}$$

Likelihood

- Observation model and encoder:

$$p_\theta(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t | x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t | z_{1:t-1}, x_{1:t-1}) \quad (1)$$

$$q_\phi(z_{1:T} | x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t | z_{1:t-1}, x_{1:T}) \quad (2)$$

- Log likelihood

$$\log p(x_{1:T}) = \log \frac{p(x_{1:T}, z_{1:T})}{p(z_{1:T} | x_{1:T})} \quad (3)$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T} | x_{1:T})} \frac{q_\phi(z_{1:T} | x_{1:T})}{p(z_{1:T} | x_{1:T})} \quad (4)$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T} | x_{1:T})} + \mathbb{KL} \left(q_\phi(z_{1:T} | x_{1:T}) || p(z_{1:T} | x_{1:T}) \right) \quad (5)$$

$$\geq \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T} | x_{1:T})} = \mathcal{L}(\theta, \phi, X) \quad (6)$$

Variational Lower Bound

- Lower bound:

$$\mathcal{L}(\theta, \phi, X) = \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \left(\frac{\prod_{t=1}^{T} p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})}{\prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})} \right) \tag{7}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \left(\sum_{t=1}^{T} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) - \sum_{t=1}^{T} \log \frac{q_\phi(z_t|z_{1:t-1}, x_{1:T})}{p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})} \right) \tag{8}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) - \tag{9}$$

$$\sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL} \left(q_\phi(z_t|z_{1:t-1}, x_{1:T}) || p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1}) \right) \tag{10}$$

- The first term is the usual **reconstruction error**.
- The second term is a regularization term, summing over the time steps the average divergence between the approximate posterior distribution of the latent variable at time t, and its real distribution.
- As in vanilla VAE, the sampling over q_ϕ requires the use of the "re parametrization trick" (see [?]), for $\mathcal{L}(\theta, \phi, X)$ to be differentiable w.r.t. θ, ϕ.

Summary DVAE

General Dynamical VAEs : generative and inference models; variational lower bound

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t | x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t | z_{1:t-1}, x_{1:t-1}) \tag{11}$$

$$q_\phi(z_{1:T} | x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t | z_{1:t-1}, x_{1:T}) \tag{12}$$

$$\begin{aligned} \mathcal{L}(\theta, \phi, X) &= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t} | x_{1:T})} \log p_{\theta_x}(x_t | x_{1:t-1}, z_{1:t}) \\ &- \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1} | x_{1:T})} \mathbb{KL}\left(q_\phi(z_t | z_{1:t-1}, x_{1:T}) || p_{\theta_z}(z_t | z_{1:t-1}, x_{1:t-1})\right) \end{aligned} \tag{13}$$

Deep Kalman Filter

Deep Kalman Filter Directed Acyclic Graph (DAG):

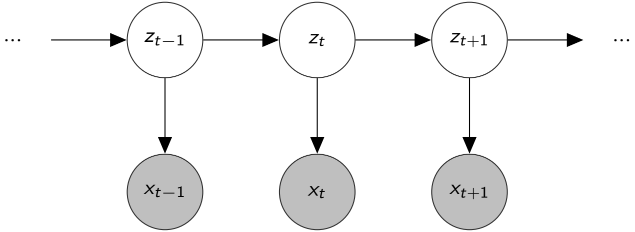


Figure: Probabilistic model of a Deep Kalman Filter

Deep Kalman Filter - generative model

Using **D-separation** on the DAG to simplify the general Dynamical Variational Auto Encoder (DVAE) expressions 11 and 12. Conditioning on z_t and z_{t-1} drives:

$$p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) = p_{\theta_x}(x_t|z_t) \tag{14}$$

$$p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t}) = p_{\theta_z}(z_t|z_{t-1}) \tag{15}$$

$$q_{\phi}(z_t|z_{1:t-1}, x_{1:T}) = q_{\phi}(z_t|z_{t-1}, x_{t:T}) \tag{16}$$

Deep Kalman Filter - generative model - 2

We then choose Gaussian distributions for $p_{\theta_x}, p_{\theta_z}$ and q_ϕ, with mean and diagonal covariance, learnt by neural networks.

$$p_{\theta_x}(x_t|z_t) = \mathcal{N}(x_t|\mu_{\theta_x}(z_t), \operatorname{diag}\sigma^2_{\theta_x}(z_t)) \tag{17}$$

$$p_{\theta_z}(z_t|z_{t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{t-1}), \operatorname{diag}\sigma^2_{\theta_z}(z_{t-1})) \tag{18}$$

$$q_\phi(z_t|z_{t-1}, x_{t:T}) = \mathcal{N}(z_t|\mu_\phi(z_{t-1}, x_{t:T}), \operatorname{diag}\sigma^2_{\theta_z}(z_{t-1}, x_{t:T})) \tag{19}$$

Some other formulations of the approximate posterior (encoder) are possible. For example:

$$q_\phi(z_t|z_{t-1}, x_t)$$
$$q_\phi(z_t|z_{1:t}, x_{1:t})$$
$$q_\phi(z_t|z_{1:T}, x_{1:T})$$

We have chosen 16 for the implementation, as it has the same formulation as the true posterior and respects the corresponding dependencies.

Deep Kalman Filter - ELBO

Using D-Separation, the Evidence Lower Bound (ELBO) 13 simplifies into:

$$\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \quad (20)$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_t|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \quad (21)$$

Deep Kalman Filter - summary

Deep Kalman Filter

- **generative model**

$$p_{\theta_x}(x_t|z_t) = \mathcal{N}(x_t|\mu_{\theta_x}(z_t), \operatorname{diag}\sigma^2_{\theta_x}(z_t)) \tag{22}$$

$$p_{\theta_z}(z_t|z_{t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{t-1}), \operatorname{diag}\sigma^2_{\theta_z}(z_{t-1})) \tag{23}$$

- **inference model**

$$q_\phi(z_t|z_{t-1}, x_{t:T}) = \mathcal{N}(z_t|\mu_\phi(z_{t-1}, x_{t:T}), \operatorname{diag}\sigma^2_{\theta_z}(z_{t-1}, x_{t:T})) \tag{24}$$

- **Variational Lower Bound (VLB) for training**

$$\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_t|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \tag{25}$$

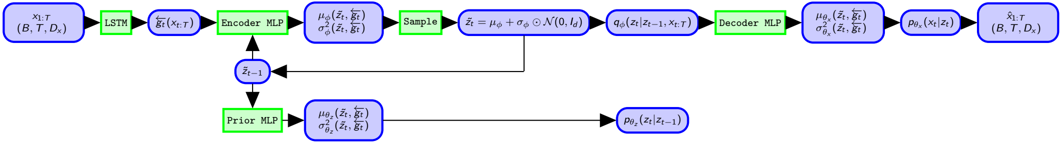
DKF - Torch

- The $\mathbb{KL}\left(q_\phi || p_{\theta_z}\right)$'s have a close form, as the two distributions are Gaussians (see **??**)
- Following [**?**], we use forward Long Short Term Memory (LSTM) to encode sequences such as $x_{1:t}$, and backward LSTM to encode sequences such as $x_{t:T}$, as inputs into the Multi Layer Perceptron (MLP) parametrizing the distributions.
- For example:

$$
\begin{aligned}
\overleftarrow{g_t} &= \text{Backward LSTM}(\overleftarrow{g_{t+1}}, x_t) \text{ (encodes } x_{t:T}) \\
q_\phi(z_t|z_{t-1}, x_{t:T}) &= \mathcal{N}(z_t|\mu_\phi(z_{t-1}, \overleftarrow{g_t}), \operatorname{diag}\sigma^2_\phi(z_{t-1}, \overleftarrow{g_t}))
\end{aligned}
$$

DKF - Torch - Schematic blocks

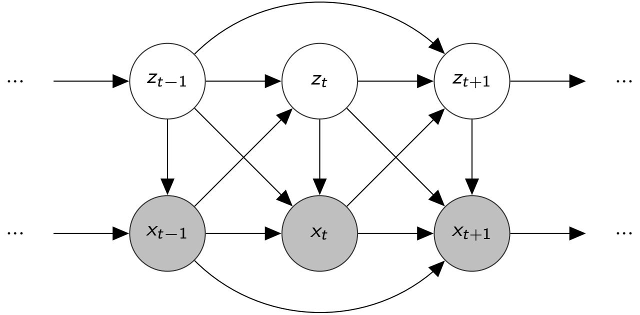
The PyTorch implementation is described below:



$$\boxed{\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^T \mathbb{E}_{q_\phi(z_t|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^T \mathbb{E}_{q_\phi(z_{t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T}) || p_{\theta_z}(z_t|z_{t-1})\right)}$$

Variational RNN

The Variational Recurrent Neural Network (VRNN) is the most expressive DVAE, in that sense that the general expressions 11, 12 and VLB 13 can not be simplified.
The Graphical Probabilistic Model (GPM) of the VRNN assumes full connections between latent variables, and between observed variables, to account for the full unsimplified expressions. Specifically:



VRNN - Summary

Variational RNN

- **generative model**

$$p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) = \mathcal{N}(x_t|\mu_{\theta_x}(x_{1:t-1}, z_{1:t}), \mathrm{diag}\, \sigma^2_{\theta_x}(x_{1:t-1}, z_{1:t})) \tag{26}$$

$$p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{1:t-1}, x_{1:t-1}), \mathrm{diag}\, \sigma^2_{\theta_z}(z_{1:t-1}, x_{1:t-1})) \tag{27}$$

- **inference model**

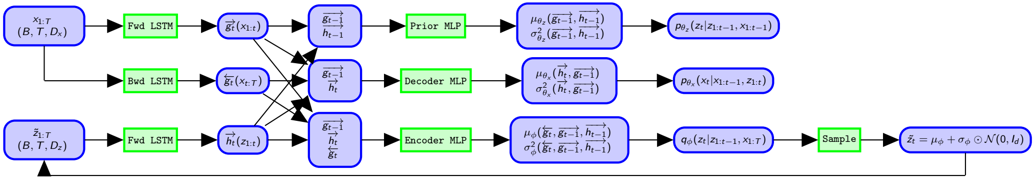
$$q_\phi(z_t|z_{1:t-1}, x_{1:T}) = \mathcal{N}(z_t|\mu_\phi(z_{1:t-1}, x_{1:T}), \mathrm{diag}\, \sigma^2_\phi(z_{1:t-1}, x_{1:T})) \tag{28}$$

- **VLB for training**

$$\begin{aligned}\mathcal{L}(\theta, \phi, X) &= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) \\ &- \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{1:t-1}, x_{1:T}) || p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})\right)\end{aligned} \tag{29}$$

VRNN - Torch

We have chosen a different implementation from [?] and used three different LSTM networks to encode $z_{1:t}$, $x_{1:t-1}$ and $x_{t:T}$ respectively.



$$\boxed{\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{1:t-1}, x_{1:T}) || p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})\right)}$$

Gaussian Process - Variational Auto Encoder

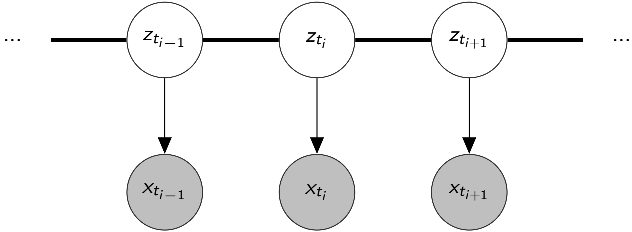


Figure: Probabilistic model of a GP-VAE

- The thick black lines indicate all latent variables follow a Gaussian Process.
- The joint distribution writes somehow differently from the one for DVAEs, as:

$$p(x_{t_1:t_T}, z_{t_1:t_T}) = p(z_{t_1:t_T})p(x_{t_1:t_T}|z_{t_1:t_T}) \tag{30}$$

$$= p(z_{t_1:t_T}) \prod_{i=1}^{T} p(x_{t_i}|x_{t_1:t_{i-1}}, z_{t_1:t_T}) \tag{31}$$

GP-VAE generative model

- **Gaussian Process prior** : the prior over the latent variables $z_{t_i} \in \mathbb{R}^L$ is a set of scalar Gaussian Process over each of the dimension $l \in \{1, ..., L\}$ of the latent variables. Formally:

$$p_{\theta_z}(z^l_{t_1:t_T}) = \mathcal{GP}(m_{\theta_z,l}(t_1 : t_T), k_{\theta_z,l}(t_1 : t_T, t_1 : t_T)) \qquad l = 1, .., L \tag{33}$$

  where the $m_{\theta_z,l}$ are the L mean functions of the Gaussian Process (GP) priors (usually chosen constant null), and the $k_{\theta_z,l}$ are the kernel functions of the GP priors.
  - by design, each of the component of the z_{t_i} is a scalar GP, with correlation over time stamps. However, the different components of a z_{t_i} are not correlated between them. The correlation across dimensions is encoded into the observation model $p_{\theta_x}(x_{t_i}|z_{t_i})$.
  - the kernels $k_{\theta_z,l}$ can be chosen differently to account for different prior knowledge of the data sequence. In [?] for example, Fortuin and al. uses a set of Gaussian Kernels with different lenghtscales.
- **Approximate posterior -encoder-** q_ϕ is a set of L Gaussian distributions of dimension T, each one accounting for a component of z_{t_i}. Formally :

$$q_\phi(z^l_{t_1:t_T}|x^l_{t_1:t_T}) = \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Sigma^l_\phi(x_{t_1:t_T})) \qquad l = 1, .., L \tag{34}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Lambda^l_\phi(x_{t_1:t_T})^{-1}) \tag{35}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), L^l_\phi(x_{t_1:t_T})L^l_\phi(x_{t_1:t_T})^T) \tag{36}$$

  where we have made explicit the different ways of defining the multivariate normal distribution, with its covariance matrix Σ^l_ϕ, its precision matrix Λ^l_ϕ, or with a Cholesky decomposition $L^l_\phi L^l_\phi$.

GP-VAE likelihood and ELBO

$$\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^{T} \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T})||p_{\theta_z}(z_{t_1:t_T})\right) \tag{37}$$

We note that:

- the $\mathbb{KL}$-divergence is actually the sum of the L $\mathbb{KL}$-divergences $\mathbb{KL}\left(q_\phi^l||p_{\theta_z}^l\right)$, which have a close form solution as both distributions are Gaussian. (see the well-known result **??**)
- the reconstruction loss term requires sampling from $q_\phi(z_{t_i}|x_{t_1:t_T})$ using the reparameterization trick as usual.
- the GP priors $p_{\theta_z}(z_{t_1:t_T})$ depend only on the time stamps $t_1, ...t_T$. If the kernel parameters are fixed -such as in [**?**]- then the priors can be computed before the training loop. If the kernel parameters are learnt with the weights of the neural nets (such as in [**?**]), then the computation must occur at each training iteration.

GP-VAE Summary

Gaussian Process VAEs

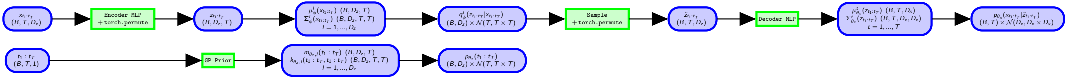
$$p(x_{t_1:t_T}, z_{t_1:t_T}) = p(z_{t_1:t_T}) \prod_{i=1}^{T} p(x_{t_i}|z_{t_i}) \tag{38}$$

$$p_{\theta_z}(z^l_{t_1:t_T}) = \mathcal{GP}(m_{\theta_z,l}(t_1 : t_T), k_{\theta_z,l}(t_1 : t_T)) \qquad l = 1, .., L \tag{39}$$

$$\begin{aligned} q_\phi(z^l_{t_1:t_T}|x^l_{t_1:t_T}) &= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Sigma^l_\phi(x_{t_1:t_T})) \qquad l = 1, .., L & (40)\\ &= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Lambda^l_\phi(x_{t_1:t_T})^{-1}) & (41)\\ &= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), L^l_\phi(x_{t_1:t_T}) L^l_\phi(x_{t_1:t_T})^T) & (42) \end{aligned}$$

$$\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^{T} \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) || p_{\theta_z}(z_{t_1:t_T})\right) \tag{43}$$

GP-VAE - Torch



$$\boxed{\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^T \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T})||p_{\theta_z}(z_{t_1:t_T})\right)}$$

Stochastic calculus survival kit - Stochastic process

Definition

A **stochastic process** is defined as:

$$X = (\Omega, \mathcal{F}, (X_t)_{t \in T}, \mathbb{P}) \tag{44}$$

$$= (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in T}, (X_t)_{t \in T}, \mathbb{P}) \tag{45}$$

where:

- Ω is a set (universe of possibles).
- $\mathcal{F}$ is a σ-algebra of parts of Ω
- $\mathbb{P}$ is a probability measure on $(\Omega, \mathcal{F})$
- $T \subset \mathbb{R}_+$ represents time
- $(\mathcal{F}_t)_{t \in T}$ is a **filtration**, ie an increasing family of sub-σ-algebras of $\mathcal{F}$ indexed by t : $\forall 0 \leq s \leq t \in T$, $\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$.
- $(X_t)_{t \in T}$ is a family of RV defined on $(\Omega, \mathcal{F})$ with values in a measurable space $(E, \mathcal{E})$ or more simply $(E, \mathcal{B}(E))$ (set E endowed with its Borelian σ-algebra).
- $(X_t)_{t \in T}$ is assumed **adapted to the filtration** $(\mathcal{F}_t)_{t \in T}$, meaning $\forall t \in T$, X_t is $\mathcal{F}_t$-measurable

Stochastic calculus survival kit - Brownian motion

Definition

A stochastic process $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ with values in $\mathbb{R}^d$ is called **Brownian motion** iff:

- $B_0 = 0$ $\mathbb{P}$-a.s.
- $\forall 0 \leq s \leq t$, the random variable $B_t - B_s$ is independent from $\mathcal{F}_t$.
- $\forall 0 \leq s \leq t$, $B_t - B_s \sim \mathcal{N}(0, Q(t-s))$
- B is continuous [a]

where the matrix $Q \in \mathbb{S}_d^{++}$ is called the **diffusion matrix**.

[a] or more exactly there exists a continuous version of B, see [?]

A core result is that the quadratic variation of the Brownian motion over an interval $[s, t]$ (equiped with a subdivison $\pi = \{s = t_0 < t_1 < ... < t_k < ... < t_n = t\}$), and defined as the limit when $|\pi| \to 0$ of $V_\pi^{(2)} = \sum_{k=0}^{n-1} |f(t_{k+1}) - f(t_k)|^2$, is:

$$\lim_{|\pi|\to 0} V_\pi^{(2)} = Q(t-s) \text{ in } L^2 \quad (46)$$

Stochastic calculus survival kit - Stochastic Integrals

Ito then proceeds to define **stochastic integrals**, starting with elementary processes:

Definition

A stochastic process $X = (X_s)_{s \in [a,b]}$ is called **elementary** if there exists a subdivision $a = t_0 < t_1 < ... < t_n = b$ of $[a, b]$, such that:

$$\forall t \in [a, b], \forall \omega \in \Omega, X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega) \mathbf{1}_{[t_i, t_{i+1}[}(t)$$

with $\forall i \in \{0, 1, .., n-1\}, X_i$ is $\mathcal{F}_{t_i}$-measurable.
This means that, in each interval $[t_i, t_{i+1}[$, $X_t(\omega)$ is independent of t and $X_t(\omega) = X_i(\omega)$.
We define $\mathcal{E}$ (resp. $\mathcal{E}_n, n > 0$) the set of all elementary processes on $[a, b]$ (resp. the subset of the $X \in \mathcal{E}$) such that all X_i have a finite moment $\mathbb{E}X_i < \infty$ (resp $\mathbb{E}(|X_i|^n) < \infty$).

Stochastic calculus survival kit - Stochastic Integrals 2

Definition

Let $X \in \mathcal{E}$, ie

$$X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega)\mathbf{1}_{[t_i, t_{i+1}[}(t)$$

The stochastic integral of X is the real random variable :

$$\int_a^b X_t dB_t := \sum_{i=0}^{n-1} X_i(B_{t_{i+1}} - B_{t_i})$$

The notion is then extended to other stochastic processes (in spaces of square integrable processes, see the annex).

Stochastic calculus survival kit - Ito's process

Definition

A process $X = (X_t)_{t\in[0,T]}$ is called a **Ito's process** if it can be written as:

$$X_t = X_0 + \int_0^t a_s ds + \int_0^t b_s dB_s \;\; \forall t \in [0, T] \tag{47}$$

where a and b are two stochastic processes such that the integrals exist (ie $a \in \Lambda^1$ and $b \in \Lambda^2$). Equivalently, we write X_t as the soltuion to the **Stochastic Differential Equation**:

$$dX_t = a_t dt + b_t dB_t$$

Stochastic calculus survival kit - Ito's formula

Theorem

An Itô's process remains an Itô's process when it is transformed by a deterministic function that is "smooth enough".
Let X be a Itô's process on $[0, T]$: $dX_t = a_t dt + b_t dB_t$.
Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x, t) \mapsto f(x, t)$ be $\mathcal{C}^{2,1}$: $\mathcal{C}^2$ in x, and $\mathcal{C}^1$ in t.
Then $(f(X_t, t))_{t \in [0,T]}$ is also an Itô's process and:

$$d\left(f(X_t, t)\right) = \frac{\partial f}{\partial t}(X_t, t)dt + \frac{\partial f}{\partial x}(X_t, t)dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(X_t, t)b_t^2 dt \tag{48}$$

The last term is Itô's complementary term.
In dimension $d > 1$:

$$d\left(f(X_t, t)\right) = \frac{\partial f}{\partial t}(X_t, t)dt + (\nabla f)^T(X_t, t)dX_t + \frac{1}{2}Tr\left((\nabla\nabla^T f)dX_t dX_t^T\right) \tag{49}$$

Beyond linear SDEs and Gaussian Processes

Beyond

Outro

Conclusions

Annexes

Annexes