Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

To cite this article: Jung-Su Ha et al J. Stat. Mech. (2019) 124008

View the article online for updates and enhancements.
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

Jung-Su Ha, Young-Jin Park, Hyeok-Joo Chae, Soon-Seo Park and Han-Lim Choi

Department of Aerospace Engineering & KI for Robotics, KAIST, Daejeon 305-701, Republic of Korea
E-mail: jsha@lics.kaist.ac.kr, yjpark@lics.kaist.ac.kr, hjchae@lics.kaist.ac.kr, sspark@lics.kaist.ac.kr and hanlimc@kaist.ac.kr

Received 31 May 2019
Accepted for publication 26 June 2019
Published 20 December 2019

Online at stacks.iop.org/JSTAT/2019/124008
https://doi.org/10.1088/1742-5468/ab3455

Abstract. We present a representation learning algorithm that learns a low-dimensional latent dynamical system from high-dimensional raw data, e.g. video. The framework builds upon recent advances in amortized inference methods that use both an inference network and a refinement procedure to output samples from a variational distribution given an observation sequence, and takes advantage of the duality between control and inference to approximately solve the intractable inference problem using the path integral control approach. The learned dynamical model can be used to predict and plan the future states; we also present the efficient planning method that exploits the learned low-dimensional latent dynamics. Numerical experiments show that the proposed path-integral control based variational inference method leads to tighter lower bounds in statistical model learning of sequential data. The supplementary video (https://youtu.be/xCp35crUoLQ) and the implementation code (https://github.com/yjparkLiCS/18-NIPS-APIAE) are available online.

Keywords: machine learning, deep learning

1 Author to whom any correspondence should be addressed.

1. Introduction

Unsupervised learning of the underlying dynamics of sequential high-dimensional sensory inputs is the essence of intelligence, because the agent should utilize the learned dynamical model to predict and plan the future state. Such learning problems are formulated as latent or generative model learning assuming that observations were emerged from the low-dimensional latent states, which includes an intractable posterior inference of latent states for given input data. In the amortized variational inference framework, an inference network is introduced to output variational parameters of an approximate posterior distribution. This allows for a fast approximate inference procedure and efficient end-to-end training of the generative and inference networks when the learning signals from a loss function are back-propagated into the inference network with reparameterization trick (Kingma and Welling 2014, Rezende et al 2014). The learning procedure is based on optimization of a surrogate loss, a lower-bound of data likelihood, which results in two source of sub-optimality: an approximation gap and an amortization gap (Cremer et al 2018, Krishnan et al 2018); the former comes from the sub-optimality of variational approximation (the gap between true posterior and optimal variational distribution) and the latter is caused by the amortized approximation.
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

(Adaptive path-integral autoencoder: representation learning and planning for dynamical systems)
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

model that explains the given observations well. If a model is parameterized with \( \theta \), the problem is formulated as a maximum likelihood estimation (MLE) problem:

\[
\theta^* = \arg\max_\theta \sum_i \log p_\theta(x_i^{(t)}),
\]

In 2. In this work, the observations are assumed to be emerged from a latent dynamical system, where a latent state trajectory, \( z_{[0,T]} \equiv \{z(t); \forall t \in [0, T]\} \), lies on a (possibly low-dimensional) latent space, \( Z \subset \mathbb{R}^d \):

\[
p_\theta(x_{1:K}) = \int p_\theta(x_{1:K}|z_{[0,T]})dp_\theta(z_{[0,T]}),
\]

where \( p_\theta(x_{1:K}|z_{[0,T]}) \) and \( p_\theta(z_{[0,T]}) \) are called a conditional likelihood and a prior distribution, respectively. In particular, we consider the state space model where latent states are governed by a continuous-time stochastic differential equation (SDE), i.e. the prior \( p_\theta(z_{[0,T]}) \) is a probability measure of a following system:

\[
dz(t) = f(z(t))dt + \sigma(z(t))dw(t), \quad z(0) \sim p_0(\cdot),
\]

where \( w(t) \) is a \( d_\sigma \)-dimensional Wiener process. Additionally, a conditional likelihood of sequential observations is assumed to be factorized along the time axis:

\[
p_\theta(x_{1:K}|z_{[0,T]}) = \prod_{k=1}^K p_\theta(x_k|z(t_k)),
\]

where \( \{t_k\} \) is a sequence of discrete time points with \( t_1 = 0, t_K = T \).

### 2.2. Amortized variational inference and multi-sample objectives

The objective function (1) cannot be optimized directly because it contains the intractable integration. To circumvent the intractable inference, a variational distribution \( q(\cdot) \) is introduced and then a surrogate loss function \( \mathcal{L}(q, \theta; x) \), which is called the evidence lower bound (ELBO), can be considered alternatively:

\[
\log p_\theta(x) = \log \int p_\theta(x|z)p_\theta(z)dz \geq \mathbb{E}_{q(z)} \left[ \log \frac{p_\theta(x|z)p_\theta(z)}{q(z)} \right] = \mathcal{L}(q, \theta; x),
\]

where \( q(\cdot) \) can be any probabilistic distribution over \( Z \) of which support includes that of \( p_\theta(\cdot) \). The gap between the log-likelihood and the ELBO is the Kullback–Leibler (KL) divergence between \( q(z) \) and the posterior \( p_\theta(z|x) \):

\[
\log p_\theta(x) - \mathcal{L}(q, \theta; x) = D_{\text{KL}}(q(z)||p_\theta(z|x)).
\]

In particular, the amortized variational inference approach introduces a conditional variational distribution, \( z \sim q_\phi(\cdot|x) \), to approximate the intractable posterior distribution. The variational distribution \( q_\phi(\cdot|x) \), which is referred to as the inference network,

\[2\] Because each observation trajectory can be considered independently, we leave trajectory index, \( i \), out and restrict our discussion to one trajectory for the sake of notational simplicity.

https://doi.org/10.1088/1742-5468/ab3455
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

is parameterized by \( \phi \), so \( \theta \) and \( \phi \) can be simultaneously updated with \( \nabla_{(\theta, \phi)} \mathcal{L}(q_\theta, \theta; \mathbf{x}) \) using the stochastic gradient ascent. Variational autoencoders (VAEs) (Kingma and Welling 2014, Rezende et al. 2014) make \( q_\theta(\cdot|\mathbf{x}) \) a reparameterizable distribution, where \( \mathbf{z} = g_\phi(\mathbf{x}, \epsilon) \) is a differentiable deterministic function of an observation \( \mathbf{x} \) and \( \epsilon \sim d(\cdot) \) sampled from a known base distribution \( d(\cdot) \). The true posterior

above procedure is referred to as importance weighted autoencoders (IWAEs) (Welling 2014, Rezende et al. 2016, Krishnan et al. 2017). The approximation gap comes up by using the variational distribution \( q_\theta(\cdot|\mathbf{x}) \) which generally yields a low variance estimator.

A tighter lower bound is achieved by using multiple samples, \( \mathbf{z}^{1:L} \), independently sampled from \( q_\theta \):

\[
\mathcal{L}^L \equiv \mathbb{E}_{\mathbf{z}^{1:L} \sim q_\theta(\cdot|\mathbf{x})} \left[ \log \frac{1}{L} \sum_{l=1}^L \frac{p_\theta(\mathbf{x}, \mathbf{z}^l)}{q_\theta(\mathbf{z}^l|\mathbf{x})} \right].
\]

It is proven that, as \( L \) increases, the bounds get tighter, i.e. \( \log p_\theta(\mathbf{x}) \geq \cdots \geq \mathcal{L}^{L+1} \geq \mathcal{L}^L \geq \cdots \), and the gap eventually vanishes (Burda et al. 2016, Cremer et al. 2017). This multi-sample objective (7) is in the class of Monte Carlo objectives (MCO) in the sense that it utilizes independent samples to estimate the marginal likelihood (Mnih and Rezende 2016), \( \hat{p}_\theta(\mathbf{x}) = \frac{1}{L} \sum_{l=1}^L \frac{p_\theta(\mathbf{z}^l)}{q_\theta(\mathbf{z}^l|\mathbf{x})} \), \( \mathbf{z}^l \sim q_\theta(\cdot|\mathbf{x}) \). Defining \( w_{\theta, \phi}(\mathbf{x}, \mathbf{z}^l) = \frac{p_\theta(\mathbf{z}^l)}{q_\theta(\mathbf{z}^l|\mathbf{x})} \), the gradient of (7) is given by:

\[
\nabla_{(\theta, \phi)} \mathcal{L}^L = \mathbb{E}_{\mathbf{z}^{1:L} \sim d(\cdot)} \left[ \sum_{l=1}^L \hat{w}^l \nabla_{(\theta, \phi)} \log w_{\theta, \phi}(\mathbf{x}, \mathbf{g}_\phi(\mathbf{x}, \epsilon^l)) \right].
\]

Since the parameter update is averaged over multiple samples with the weights \( \hat{w}^l \), the above procedure is referred to as importance weighted autoencoders (IWAEs) (Burda et al. 2016). The performance of IWAE’s training crucially depends on the variance of the importance weights \( \hat{w} \) (or equivalently, on the effective sample size), which can be reduced by (i) increasing the number of samples and (ii) decreasing the gap between the proposal and the true posterior distribution; when the proposal \( q_\theta(\cdot|\mathbf{x}) \) is equal to the true posterior \( p_\theta(\cdot|\mathbf{x}) \), the variance is reduced to 0, i.e. \( \hat{w}^l = 1/L \).

### 2.3. Semi-amortized variational inference with iterative refinement

As mentioned previously, the performance of generative model learning depends on the gap between the variational and the posterior distributions. Thus, the amortized inference has two sources of this gap: the approximation and amortization gaps (Cremer et al. 2018, Krishnan et al. 2018). The approximation gap comes up by using the variational distribution to approximate the posterior distribution, which is given by the KL-divergence between the posterior distribution and the optimal variational distribution. The amortization gap is caused by the limit of the expressive power of inference networks, where the variational parameters are not individually optimized for each observation but amortized over entire observations. To address the issue of the amortization gap, a hybrid approach can be considered; for each observation, the variational distribution is refined individually from the output of the inference network. Compared to the amortized variational inference, this hybrid approach, coined
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems

3. Path integral adaptation for variational inference

3.1. Controlled SDE as variational distribution and structured inference network

When handling sequential observations, the variational distribution family should be carefully chosen so as to efficiently handle increasing dimensions of variables along the time-axis. In this work, the variational proposal distribution is given by the trajectory distribution of a controlled stochastic dynamical system, where the controls, \( \mathbf{u} \in \mathbb{R}^{d_u} \), and parameters of an initial state distribution, \( q_0 \), serve as variational parameters, i.e. the proposal \( q_u(\mathbf{z}(0:T)) \) is a probability measure of a following system:

\[
d\mathbf{z}(t) = f(\mathbf{z}(t))dt + \sigma(\mathbf{z}(t))(\mathbf{u}(t)dt + d\mathbf{w}(t)), \quad \mathbf{z}(0) \sim q_0(\cdot). \tag{9}
\]

By applying Girsanov’s theorem in appendix A (see appendix A in the supplementary information, available online at stacks.iop.org/JSTAT/19/124008/mmedia) that provides the likelihood ratio between \( p(\mathbf{z}(0:T)) \) and \( q_u(\mathbf{z}(0:T)) \), the ELBO is written as:

\[
\mathcal{L} = \mathbb{E}_{q_u(\mathbf{z}(0:T))} \left[ \log p_\theta(\mathbf{x}_{1:T} | \mathbf{z}(0:T)) + \log \frac{p_\theta(z(0))}{q_0(z(0))} - \frac{1}{2} \int_0^T ||\mathbf{u}(t)||^2 dt - \int_0^T \mathbf{u}(t)^T d\mathbf{w}(t) \right]. \tag{10}
\]

Then, the problem of finding the optimal variational parameters \( \mathbf{u}^* \) and \( q_0^* \) (or equivalently, the best approximate posterior) can be formulated as a SOC problem:

\[
\mathbf{u}^*, q_0^* = \arg\min_{\mathbf{u}, q_0} \mathbb{E}_{q_u(\mathbf{z}(0:T))} \left[ V(\mathbf{z}(0:T)) + \frac{1}{2} \int_0^T ||\mathbf{u}(t)||^2 dt + \int_0^T \mathbf{u}(t)^T d\mathbf{w}(t) \right], \tag{SOC}
\]

where \( V(\mathbf{z}(0:T)) \equiv -\log \frac{p_\theta(z(0))}{q_0(z(0))} - \sum_{k=1}^K \log p_\theta(\mathbf{x}_k | \mathbf{z}(t_k)) \) serves as a state cost of the SOC problem.

Suppose that the control policy is discretized along the time-axis with the control parameters \( \{\mathbf{u}_k^{ff}, \mathbf{K}_k\}_{k=1,...,K-1} \) as \( \mathbf{u}(t, \mathbf{z}(t)) = \mathbf{u}_k^{ff} - \mathbf{K}_k \mathbf{z}(t), \quad \forall t \in [t_k, t_{k+1}) \), and the initial distribution is modeled to be the Gaussian distribution, \( q_0(\cdot) = \mathcal{N}(\cdot; \hat{\mu}_0, \Sigma_0) \). Once the inference problem is converted into the SOC problem, the principle of optimality (Bellman 2013) provides the sophisticated and efficient structure of inference networks. Note that, by the principle of optimality, the optimal initial state distribution depends on the cost for all time horizon \([0, T]\) but the optimal control policy at \( t \) only relies on the future cost in \([t, T]\). Such a structure can be implemented using a backward recurrent neural network (RNN) to output the approximate optimal control policy; while the hidden states of the backward RNN compress the information of a given observation sequence backward in time, the hidden state at each time step, \( k = K - 1, ..., 2 \), outputs the control policy parameters, \( \{\mathbf{u}_k^{ff}, \mathbf{K}_k\} \). Finally, the first hidden state additionally outputs the initial distribution parameters, \( \{\hat{\mu}_0, \Sigma_0, \mathbf{u}_1^{ff}, \mathbf{K}_1\} \). For the detailed descriptions and illustrations, see figure 3 and algorithm 2 in appendix C.

https://doi.org/10.1088/1742-5468/ab3455
3.2. Adaptive path-integral autoencoder

(SOC) is in a class of linearly-solvable optimal control problems (Todorov 2009) of which the objective function can be written as a KL-divergence form:

$$J = D_{KL}(q_u(z_{[0,T]})\|p^*(z_{[0,T]})) - \log \xi,$$

(11)

where $p^*$, represented as $dp^*(z_{[0,T]}) = \exp(-V(z_{[0,T]}))dp_\theta(z_{[0,T]})/\xi$, is a probability measure induced by optimally-controlled trajectories and $\xi \equiv \int \exp(-V(z_{[0,T]})dp_\theta(z_{[0,T]})$ is a normalization constant (see appendix A for details). By applying Girsanov’s theorem again, the optimal trajectory distribution is expressed as:

$$dp^*(z_{[0,T]}) \propto dq_u(z_{[0,T]}) \exp(-S_u(z_{[0,T]})),$$

(12)

$$S_u(z_{[0,T]} = V(z_{[0,T]} + \frac{1}{2} \int_0^T ||u(t)||^2 dt + \int_0^T u(t)^T dw(t).$$

(13)

This implies that the optimal trajectory distribution can be approximated by sampling a set of trajectories according to the controlled dynamics with $u(t)$, i.e. $z'_{[0,T]} \sim q_u(\cdot)$, and assigning their importance weights as $\tilde{w}^l = \frac{\exp(-S_u(z^l_{[0,T]}))}{\sum_{i=1}^L \exp(-S_u(z^l_{[0,T]}))}, \forall l \in \{1, \ldots, L\}$. Similar to the MCO’s case, the variance of importance weights decreases as the control input $u(\cdot)$ gets closer to the true optimal control input $u^*(\cdot)$ and it reduces to 0 when $u(t) = u^*(t,z(t))$ (Thijssen and Kappen 2015).

The path-integral control is a sampling-based SOC method, which approximates the optimal trajectory distribution, $\hat{p}^*$, with weighted sample trajectories using (12) and (13) and updates control parameters based on moment matching of $q_u$ to $\hat{p}^*$. Suppose that $\hat{p}^*$ is approximated with sample trajectories and their weights, $\{z^l_{[0,T]}, \tilde{w}^l\}_{l=1}^L$, as above and let $u^{ff}(t)$ and $K(t)$ represent feedforward control and feedback gain, respectively. This work considers a standardized linear feedback controller to regularize the first and second moments of trajectory distributions, where a control input has a form as:

$$u(t) = u^{ff}(t) + K(t)\Sigma^{-1/2}(t)(z(t) - \mu(t)),$$

(14)

where $\mu(t) = \sum_{l=1}^L \tilde{w}^l z^l(t)$ and $\Sigma(t) = \sum_{l=1}^L \tilde{w}^l(z^l(t) - \mu(t))(z^l(t) - \mu(t))^T$ are the mean and covariance of the state w.r.t. $\hat{p}^*$, respectively. Suppose a new set of trajectories and their weights is obtained by a (previous) control policy $u(t) = \bar{u}^{ff}(t) + \bar{K}(t)\Sigma^{-1/2}(t)(z(t) - \bar{\mu}(t))$. Then, the path integral control theorem in appendix B gives the update rules as:

$$u^{ff}(t) dt = \bar{u}^{ff}(t) dt + \bar{K}(t)\Sigma^{-1/2}(t)(\mu(t) - \bar{\mu}(t)) dt + \eta \sum_{l=1}^L \tilde{w}^l dw^l(t),$$

(15)

$$K(t) dt = \bar{K}(t)\Sigma^{-1/2}(t)\Sigma^{1/2}(t) dt + \eta \sum_{l=1}^L \tilde{w}^l dw^l(t) (\Sigma^{-1/2}(t)(z^l(t) - \mu(t)))^T,$$

(16)

with the adaptation rate $\eta$. The initial state distribution also can be updated into $q_0(\cdot) = \mathcal{N}(\cdot; \bar{\mu}_0, \bar{\Sigma}_0)$.
\begin{equation}
\hat{\mu}_0 = \sum_{l=1}^{L} \tilde{w}_l z_l(0), \quad \hat{\Sigma}_0 = \sum_{l=1}^{L} \tilde{w}_l (z_l(0) - \hat{\mu}_0)(z_l(0) - \hat{\mu}_0)^T.
\end{equation}

Starting from the variational parameters, \{\hat{\mu}_0, \hat{\Sigma}_0, u_{1:K-1}, K_{1:K-1}\}, given by the inference network and \bar{\mu}(t) = 0, \bar{\Sigma}(t) = I, the update rules in (15)–(17) gradually refine the parameters of \(q_u\) in order for the resulting trajectory distribution to be close to the posterior distribution. After \(R\) adaptations, the MCO and its gradient are estimated by:

\begin{equation}
\hat{L}_L = \log \frac{1}{L} \sum_{l=1}^{L} \exp(-S_u(z_{[0,T]})), \quad \nabla_{\theta, \phi} \hat{L}_L = - \sum_{l=1}^{L} \tilde{w}_l \nabla_{\theta, \phi} S_u(z_{[0,T]}),
\end{equation}

where \(\theta\) and \(\phi\) denote the parameters of the generative model, i.e. \(f(z), \sigma(z), p_0(z)\) and \(p(x|z)\), and the inference network, i.e. the backward RNN, respectively. Because all procedures in the path integral adaptation and MCO construction are differentiable, they can be implemented by a fully differentiable network with \(R\) recurrences, which we named adaptive path integral autoencoder (APIAE).

Note that the inference, reconstruction, and gradient backpropagation of APIAE can operate independently for each of \(L\) samples. Consequently, the computational cost grows linearly with the number of samples, \(L\), and the number of adaptations, \(R\). As implemented in IWAE (Burda et al 2016), we replicated each observation data \(L\) times and the whole operations were parallelized with GPU. We implemented APIAE with Tensorflow (Abadi et al 2016); the pseudo code and algorithmic details of APIAE are given in the appendix C.

4. High-dimensional motion planning with learned latent model

High-dimensional motion planning is a challenging problem because of the curse of dimensionality: the size of the configuration space exponentially increases with the number of dimensions. However, like in the latent variable model learning, it might be a reasonable assumption that configurations a planning algorithm really needs to consider form some sort of low-dimensional manifold in the configuration space (Vernaza and Lee 2012), and the learned generative model provides stochastic dynamics in that manifold. Once this low-dimensional representation is obtained, any motion planning algorithm can solve high-dimensional planning problem very efficiently by utilizing it to restrict the search space.

More formally, suppose that the initial configuration, \(x_1\), and corresponding latent state, \(z(0)\), are given and the cost function, \(C_k(x_k)\), encodes given task specifications of a planning problem, e.g. desirability/undesirability of certain configurations, a penalty for obstacle collision, etc. Then, the planning problem can be converted into the problem of finding the optimal trajectory distribution, \(q_u\), that minimizes the following objective function:

\begin{equation}
J(q_u) = \mathbb{E}_{x_1:K \sim p_0(\cdot|z_{[0,T]}), z_{[0,T]} \sim q_u(\cdot)} \left[ \sum_{k=1}^{K} C_k(x_k) + D_{KL}(q_u(z_{[0,T]}) || p_\theta(z_{[0,T]})) \right].
\end{equation}

https://doi.org/10.1088/1742-5468/ab3455

8
That is, we want to find parameters, \( u \), of the trajectory distribution which not only is likely to generate sample configuration sequences achieving the lower planning cost but also does not deviate a lot from the (learned) prior, \( p(\mathbf{z}_{0:T}) \). The solution can be found using the aforementioned adaptive path integral control method, where its state cost function is set as:

\[
V(\mathbf{z}_{0:T}) \equiv E_{p(\mathbf{x}_{1:K}|\mathbf{z}_{0:T})} \left[ \sum_{k=1}^{K} C_k(\mathbf{x}_k) \right]
\]

and the initial state distribution is not updated in the adaptation process. After the adaptations with this state cost function, the resulting plan can simply be sampled from the generative model, e.g. \( \mathbf{x}_{1:K} \sim p_{\theta}(\cdot|\mu_{0:T}) \). Note that the time interval \( t_k - t_{k-1} \) and the trajectory length \( K \) can differ in the training and planning phases because continuous-time dynamics is dealt with.

5. Related work

To address the complexity raised from temporal structures of data, several approaches that build a sophisticated approximate inference model have been proposed. For example, Karl et al (2017) used the locally linear latent dynamics by introducing transition parameters, where an inference model infers transition parameters rather than latent states from the local transition. Johnson et al (2016) combined a structured graphical model in latent space with a deep generative network, where an inference network produces local evidence potentials for the message passing algorithms. Fraccaro et al (2017) constructed two layers of latent models, where linear-Gaussian dynamical systems governed two latent layers and the observation at each time step was related to the middle layer independently; the inference model in this framework consists of independent VAE’s inference networks at each time-step and the Kalman smoothing algorithm along the time axis. Finally, deep Kalman smoother (DKS) in Krishnan et al (2017) parameterized the dynamical system by a deep neural network and built an inference network as it has the same structure with the factorized posterior distribution. The idea of MCOs was also used in the temporal setting. Maddison et al (2017), Le et al (2018), Naesseth et al (2018) adapted the particle filter (PF) algorithm as their inference models and utilized a PF’s estimator of the marginal likelihood as an objective function of training which (Maddison et al 2017) named the filtering variational objectives (FIVO).

These approaches can be viewed as attempts to reduce the approximation gap; by building the inference model in sophisticated ways that exploit underlying structure of data, the resulting variational family could flexibly approximate the posterior distribution. To overcome the amortization gap caused by inference networks, the semi-amortized method utilizes an iterative refinement procedure for improving variational distribution. Let \( q_\phi \) and \( q^{*} \) be the variational distributions from the inference network and from the refinement procedure, i.e. before and after the refinement, respectively. Hjelm et al (2016) adopted adaptive importance sampling to refine the variational parameters, and the generative and inference networks are trained separately with \( \nabla_\theta L(q^*, \theta; \mathbf{x}) \) and \( \nabla_\phi D_{KL}(q^*||q_\phi) \), respectively. Krishnan et al (2018) used stochastic variational inference as a refinement procedure, and the generative and inference networks

https://doi.org/10.1088/1742-5468/ab3455
are also trained separately with $\nabla_{\theta} \mathcal{L}(q^*, \theta; x)$ and $\nabla_{\phi} \mathcal{L}(q_{\phi}, \theta; x)$, respectively. Kim et al (2018) also used stochastic variational inference but proposed the end-to-end training by allowing the learning signals to be backpropagated into the refinement procedure, and showed this end-to-end training outperformed the separate training.

This work presents a semi-amortized variational inference method for temporal data. In summary, we parameterize the variational distribution by control input and transformed the approximate inference into the SOC problem. Our method utilizes the structured inference network based on the principle of optimality which has a similar structure to the inference network of DKS (Krishnan et al 2017). The adaptive path-integral control method, which can be viewed as adaptive importance sampling in trajectory space (Kappen and Ruiz 2016), is then adopted as a refinement procedure. Ruiz and Kappen (2017) also used the adaptive path integral approach to solve smoothing problems and showed the path integral-based smoothing method could outperform the PF-based smoothing algorithms. Finally, by observing all procedures of the path integral smoothing are differentiable, the inference and generative networks are trained in the end-to-end manner. Note that APIAE is not the first algorithm that implements an optimal planning/control algorithm into a fully-differentiable network. In Tamar et al (2016), Okada et al (2017), Karkus et al (2017), similar iterative refinement procedures were built as differentiable networks to learn solutions of control problems in an end-to-end manner; the fact that iterative methods were generally used to solve control problems can be a rationale for utilizing refinement to approximate inference for sequential data.

In addition, there is a non-probabilistic branch of representation learning of dynamical systems, e.g. Watter et al (2015), Jonschkowski and Brock (2015), Banijamali et al (2018), Lesort et al (2018). They basically stack two consecutive observations to contain the temporal information and learn the dynamical model based on a carefully designed loss function considering the stacked data as one observation. As shown in appendix D, however, when the observations are highly-noisy (or even worse, when the system is unobservable with the stacked data), stacking a small number of observations prohibits the training data from containing enough temporal information for learning rich generative models.

Lastly, there have been some recent works to utilize a low-dimensional latent model for motion planning. Chen et al (2016) exploited the idea of VAEs to embed dynamic movement primitives into the latent space. In Ha et al (2018), Gaussian process dynamical models (Wang et al 2008) served as a latent dynamical model and was utilized for planning in a similar way with this work. Though the dynamics were not considered, Ichter et al (2018), Zhang et al (2018) used the conditional VAEs to learn a non-uniform sampling methodology of a sampling-based motion planning algorithm.

6. Experiment

In our experiments, we would like to show that the proposed method is a complementary technique to the existing methods; the APIAE can play a role in constructing more expressive posterior distribution by refining the variational distribution from the
existing approximate inference methods. To support our statement, we built APIAEs upon the FIVO and IWAE frameworks and compared with the model without adaptation procedures.

We set our APIAE parameters as $L = 8$, $R = 4$, and $K = 10$ during experiments. Quantitative studies about the effect of varying these parameters are discussed in the appendix. Feedback gain is only used for the planning, since matrix inversion in (16) requires Cholesky decomposition which is often numerically unstable during the training. We would refer the readers to the appendix D and the supplementary video for more experimental details and results.

6.1. Dynamic pendulum

The first experiment addresses the system identification and planning of inverted pendulum with the raw images. The pendulum dynamics is represented by the second order differential equation for angle of the pendulum, $\psi$, as $\ddot{\psi} = -9.8 \sin(\psi) - \dot{\psi}$. We simulated the pendulum dynamics by injecting the disturbance from random initial states and then made sequences of $16 \times 16$ sized images corresponding to the pendulum state with the time interval, $\delta t = 0.1$. This set of sequence images was training data of APIAE, i.e. $x_t$ lied in 256-dimensional observation space. 3000 and 500 data are used for training and test, respectively.

Figure 1(a) shows the constructed 2-dimensional latent space; each point represents the posterior mean of the observation data and it is shown that the angle and the angular velocity are well-encoded in 2-dimensional space. As shown in figure 1(b), the learned dynamical model was able to successfully reconstruct the noisy observations, predict and plan the future images. For the planning, the cost functions were set to penalize the difference between the last image of the generated sequence and the target image in figure 1(c) to encode planning problems for swing-up, -down, -left, and -right.

6.2. Human motion capture data

The second experiment addresses a motion planning of a humanoid robot with 62-dimensional configuration space. We utilized human motion capture data from the Carnegie Mellon University motion capture (CMU mocap) database for the learning; the training data was a set of (short) locomotion, e.g. for standing, walking, and turning. The 62-dimensional configurations consist of angles of all joints, roll and pitch angles, vertical position of the root, yaw rate of the root, and horizontal velocity of the root. The global (horizontal) position and heading orientation are not encoded in the generative model (only velocities are encoded), but they can be recovered by integration when an observation sequence is given. The original data were written at 120 Hz, and we downsampled them to 20 Hz and cut them every ten time steps, i.e. $\delta t = 0.05$, $K = 10$. 1043 and 173 data are used for training and test, respectively. We utilized the DeepMind Control Suite (Tassa et al 2018) for parsing the data and visualizing the results.

Figures 2(a)–(c) illustrate the posterior mean states of the training data colored by some physical quantities of the ground truth; we can observe that (a) locomotion is basically embedded along the surface of the cylinder, while (b) they were arranged in the order of the yaw rates along the major axis of the cylinder and (c) motions with
lower forward velocities were embedded into smaller radius cycles. Also, figure 2(d) shows that APIAE successfully reconstructed the data. Compared to the pendulum example, where the Wiener process in latent dynamics models disturbance into the system and the prediction can be made simply by ignoring the disturbance, the framework in this example uses the Wiener process to model the uncertainty in human’s decision, e.g. whether to turn left or right, to increase or decrease their speed, etc, similar to the modeling of the bounded rationality (Genewein et al. 2015) or the maximum entropy IRL (Ziebart et al. 2008); as shown in figure 2(e), from the very same initial pose, the framework predicts multiple future configurations for, e.g. going straight, turning left or right (the ratio between motions eventually matches that of the training dataset) and these predictions play essential roles in the planning. We then formulated planning problems, where the cost function penalized collision with an obstacle, large yaw rate, and distance from the goal. Figures 2(f)–(g) show that the proposed method successfully generated the natural and collision-free motion toward the goal.

6.3. Quantitative results

It is easily thought that powerful inference methods via resampling or refinements make the bound tighter, but achieving a tighter bound during learning does not directly imply a better model learning (Rainforth et al. 2018). To investigate this, we have compared the lower bound, the reconstruction and prediction abilities of the models learned by the proposed and baseline algorithms. The results are reported in table 1 (higher is better). Interestingly, we can observe that learning with both the resampling and path-integral refinements resulted in the best reconstruction ability as well as the tightest bound, but the best prediction was achieved by the model learned only with the refinements. It implies that while powerful inference can lead to a tighter bound and a good reconstruction, a bias in the gradients can prevent the resulting model from being accurate (note that the gradient components from the resampling are generally

---

3 Mocap prediction is omitted, because a proper measure for the prediction is unclear.

https://doi.org/10.1088/1742-5468/ab3455
ignored because it causes high variance of the gradient estimator (Maddison et al. 2017, Le et al. 2018, Naesseth et al. 2018)). In the planning side, the prediction power is crucial because the (learned) generative model needs to sample meaningful and diverse configuration sequences. We conclude that the resampling procedure would be better to utilize only for planning, not for learning, and this also would be the same in other application domains like 3-dimensional human motion tracking, where the prediction ability is more important.

Figure 2. Mocap results. The learned latent space colored by (a) the gait phase, (b) yaw rate, and (c) forward velocity of the ground truth. We set the phase as 0 when the left feet touch the ground and as π when the right feet touch the ground. (d) Reconstruction. (e) Prediction results from the same initial poses. (f)–(g) Locomotion planning results.
7. Conclusion

In this paper, a semi-amortized variational inference method for sequential data was proposed. We parameterized a variational distribution by control input and transformed an approximate inference into a SOC problem. The proposed framework utilized the structured inference network based on the principle of optimality and adopted the adaptive path-integral control method as a refinement procedure. The experiments showed that the refinement procedure helped the learning algorithm achieve tighter lower bound. Also, it is shown that the valid dynamical model can be identified from sequential raw data and utilized to plan the future configurations.

Acknowledgments

This work was supported by the Agency for Defense Development under contract UD150047JD.

References

Banijamali E, Shu R, Ghavamzadeh M, Bui H and Ghodsi A 2018 Robust locally-linear controllable embedding Int. Conf. Artificial Intelligence and Statistics
Burda Y, Grosse R and Salakhutdinov R 2016 Importance weighted autoencoders Int. Conf. on Learning Representations
Chen N, Karl M and van der Smagt P 2016 Dynamic movement primitives in latent space of time-dependent variational autoencoders Int. Conf. on Humanoid Robots (IEEE) pp 629–36
Cremer C, Morris Q and Duvenaud D 2017 Reinterpreting importance-weighted autoencoders ICLR Workshop
Cremer C, Li X and Duvenaud D 2018 Inference suboptimality in variational autoencoders (arXiv:1801.03558)
Adaptive path-integral autoencoder: representation learning and planning for dynamical systems


Ichter B, Harrison J and Pavone M 2018 Learning sampling distributions for robot motion planning Int. Conf. on Robotics and Automation (ICRA)


Karl M, Soelch M, Bayer J and van der Smagt P 2017 Deep variational bayes filters: Unsupervised learning of state space models from raw data Int. Conf. on Learning Representations


Kingma D P and Welling M 2014 Auto-encoding variational bayes Int. Conf. on Learning Representations

Krishnan R G, Shalit U and Sontag D 2017 Structured inference networks for non-linear state space models AAAI pp 2101–9

Krishnan R G, Liang D and Hoffman M 2018 On the challenges of learning with inference networks on sparse, high-dimensional data Int. Conf. on Artificial Intelligence and Statistics

Le T A, Igl M, Jin T, Rainforth T and Wood F 2018 Auto-encoding sequential Monte Carlo Int. Conf. on Learning Representations


Mnih A and Rezende D 2016 Variational inference for monte carlo objectives Int. Conf. on Machine Learning pp 2188–96

Naesseth C A, Linderman S W, Ranganath R and Blei D M 2018 Variational sequential Monte Carlo Int. Conf. on Artificial Intelligence and Statistics


Rainforth T, Kosiorek A R, Le T A, Maddison C J, Igl M, Wood F and Teh Y W 2018 Tighter variational bounds are not necessarily better Int. Conf. on Machine Learning

Rezende D J, Mohamed S and Wierstra D 2014 Stochastic backpropagation and approximate inference in deep generative models Int. Conf. on Machine Learning pp 1278–86


Vernaza P and Lee D D 2012 Learning and exploiting low-dimensional structure for efficient holonomic motion planning in high-dimensional spaces Int. J. Robot. Res. 31 1739–60


Ziebart B D, Maas A L, Bagnell J A and Dey A K 2008 Maximum entropy inverse reinforcement learning AAAI vol 8 (Chicago, IL) pp 1433–8

https://doi.org/10.1088/1742-5468/ab3455