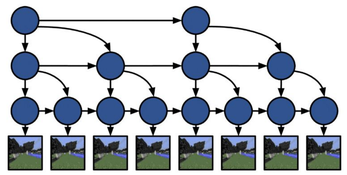
Clockwork Variational Autoencoders

Vaibhav Saxena[12], Jimmy Ba[12], Danijar Hafner[123]

[1] University of Toronto, [2] Vector Institute, [3] Google Brain

1 Introducing Clockwork VAEs

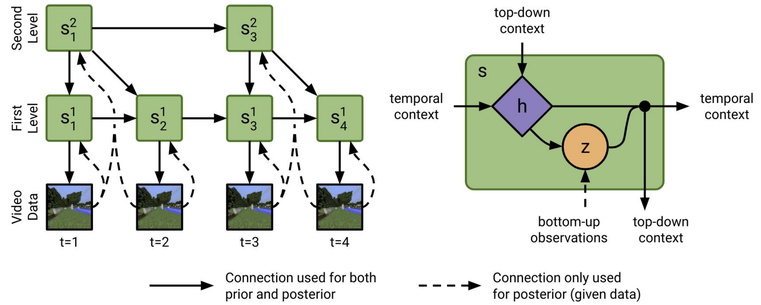
1. Clockwork VAE is a temporally-abstract recurrent latent variable models.
2. It predicts the future on multiple time-scales, preserving long-term dependencies.
3. It can predict upto 1000 frames on video datasets while preserving high-level details.
4. It can separate and store slow-changing semantics at higher levels of the hierarchy.
5. It adapts to the speed of the sequence, shifting information across slow and fast moving recurrent chains.



Temporally abstract recurrent chains in a Clockwork VAE with 3 levels and temporal abstraction factor 2.

The amount of spatial abstraction determines the number of parameters in the model. Adding temporal abstraction does not affect the model size.

2 Hierarchy of Latent Sequences



- Clockwork VAE consists of a hierarchy of recurrent latent variables, where each level transitions at a different clock speed.
- Transitions slow down exponentially as we go up in the hierarchy with a factor we call the temporal abstraction factor.
- The posterior at each level is comprised of top-down and bottom-up information, while the prior (generation) is only top-down.
- Each latent state is a combination of a deterministic and a stochastic state, with the weights determining the deterministic state shared by the posterior and prior computations.

3 Training Objective

1. We can factorize the joint distribution of a sequence of images and latents at every level into two terms: (1) the reconstruction terms of the images given their lowest level latents, and (2) state transitions at all levels conditioned on the previous latent and the latent above.

$$p(x_{1:T}, s^{1:L}_{1:T}) \doteq \Big(\prod_{t=1}^{T} p(x_t \mid s^1_t)\Big)\Big(\prod_{l=1}^{L}\prod_{t\in\mathcal{T}_l} p(s^l_t \mid s^l_{t-1}, s^{l+1}_t)\Big)$$

2. To implement this distribution and its inference model, Clockwork VAE utilizes the following components:

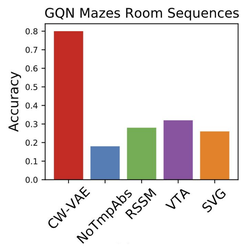
$$\begin{aligned}
&\text{Encoder:} && e^l_t = e(x_{t:t+k^{l-1}-1}) \\
&\text{Posterior transition } q^l_t\text{:} && q(s^l_t \mid s^l_{t-1}, s^{l+1}_t, e^l_t) \\
&\text{Prior transition } p^l_t\text{:} && p(s^l_t \mid s^l_{t-1}, s^{l+1}_t) \\
&\text{Decoder:} && p(x_t \mid s^1_t).
\end{aligned}$$

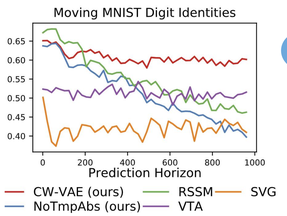
3. Because we cannot compute the likelihood of the training data under the model in closed form, we use the ELBO as our training objective. This training objective optimizes a reconstruction loss at the lowest level, and a KL regularizer at every level in the hierarchy summed across all timesteps.

$$\max_{e,q,p} \sum_{t=1}^{T} \mathrm{E}_{q^1_t}[\ln p(x_t \mid s^1_t)] - \sum_{l=1}^{L}\sum_{t\in\mathcal{T}_l} \mathrm{E}_{q^l_{t-1}q^{l+1}_t}\Big[\mathrm{KL}[q^l_t \parallel p^l_t]\Big]$$

4 Video Prediction Benchmark

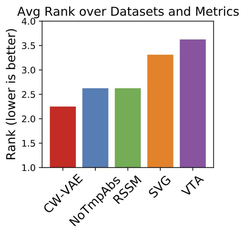
1. For GQN Maze videos, we compute the prediction accuracy for three different high-level categories of sequences of rooms: agent staying in the same room, going into the hallway, and then coming back to the original room. Video generated by Clockwork VAE greatly matches with the ground truth.





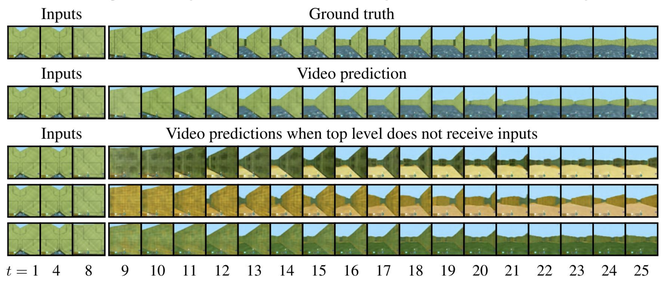
2. On Moving MNIST, Clockwork VAE can predict sequences upto **1000 timesteps** with a high accuracy of digit identities.

3. Finally, we rank the models based on their performance averaged over 4 different metrics and datasets.

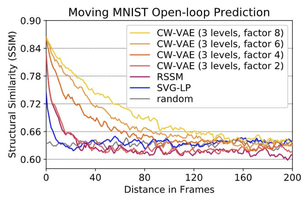


5 Semantic Content Separation

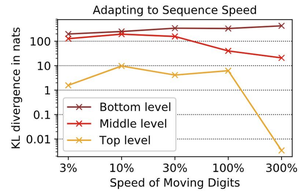
1. Generating video predictions with top-level reset to the prior.



2. Measuring video prediction accuracy when tuning the temporal abstraction factor.



3. Measuring amount of information at different levels when changing video speed.



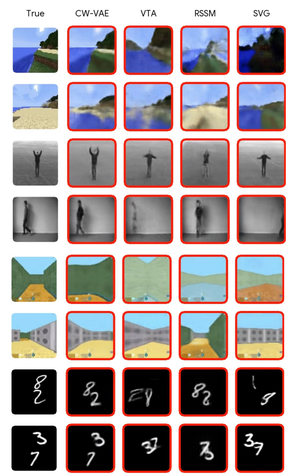
6 Find out more

Try out Clockwork VAEs!

Useful resources are available on the project website:

danijar.com/cwvae

- Minecraft dataset released with this paper.
- Video prediction examples on Minecraft, KTH Action, GQN Mazes, and Moving MNIST.
- Training and evaluation code in TensorFlow and a JAX reimplementation.
- Video generation code for level-resetting.



Open-loop video prediction for ~100 frames (10 seconds)

@danijarh

Website with videos, data, and code: danijar.com/cwvae