Attend, Infer, Repeat: Fast Scene Understanding with Generative Models

S.M. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K.Kavukcuoglu, G. E. Hinton

Google DeepMind, London, UK

NIPS 2016/ Presenter: Ji Gao

Outline

Motivation

- Generative models:

	Deep Generative Models	Structure Generative Models
+	Get good performance	Easy to interpret
-	Hard to interpret	Inference is hard, slow

- Can we create interpretable and efficient models by combining structure generative models and deep models?

- Task: Scene understanding: inference the objects in 2D or 3D scenes

Task

- Given an image x, task is to generate the description of the image z.
- Decompose to objects: z is structured in groups of variables z^i.
- Each z^i represent an attribute of a certain object. An object can be represented by a group of z^i.

Bayesian view of scene interpretation

Bayesian view of scene interpretation

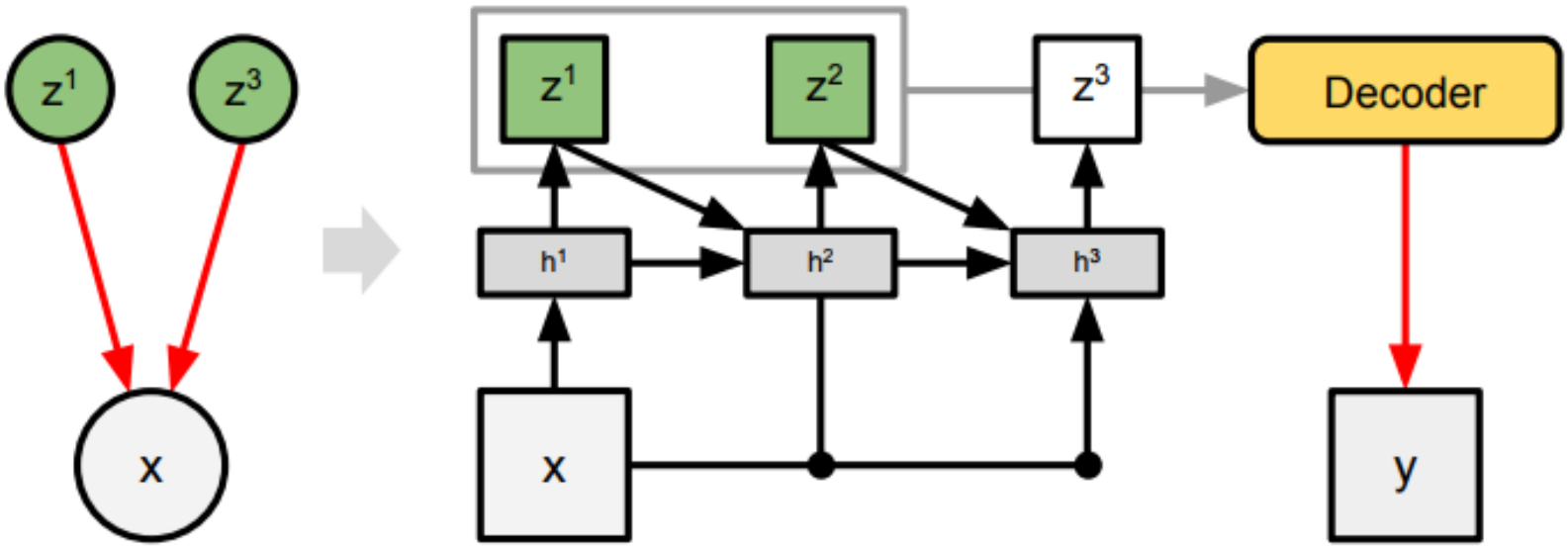
$$P_\theta(x) = \sum_{n=1}^{N} P(n) \int P_\theta(z|n) P_\theta(x|z) dz$$

n: Number of objects

z: Latent variables

θ: Parameters of the model

Overview: Inference Network and Generative Model



- Black arrow: Inference network
- Red arrow: Model
- Trained together in a feed-forward manner, without supervision.

Variational Inference

- In Bayesian equation, posterior is important but hard to compute when latent variables are included.
- Approximate the posterior $P(Z|X)$ by a variational distribution $Q(Z)$
- Minimize KL divergence

$$KL(Q||P) = \int Q(Z) \log \frac{Q(Z)}{P(Z|X)} dZ = \int Q(Z) \log \frac{Q(Z)}{P(Z,X)} dZ + logP(X$$

.

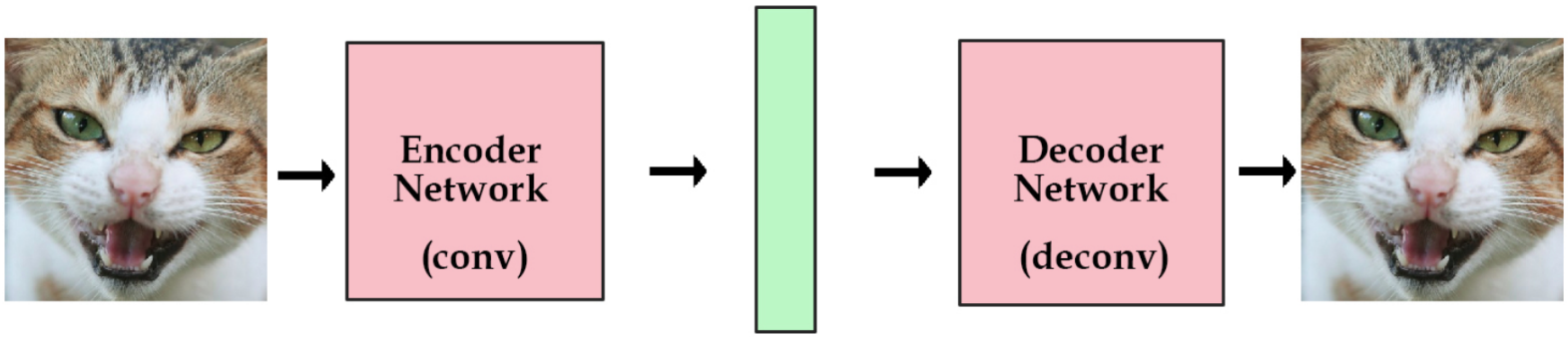
- Equivalent to maximize evidence lower bound (negative free energy)

$$ELOB(Q) = \int Q(Z) \log(P(Z,X)) dZ - \int Q(Z) \log Q(Z) dZ$$

- If Q is among the mean-field variational family, then $Q(Z) = \prod_{i=1}^{m} Q_i(z_i)$

Variational Autoencoder

- Variational autoencoder: minimize the generalization error plus a latent error



Inference

- Model:

$$P_\theta(x) = \sum_{n=1}^{N} P(n) \int P_\theta(z|n) P_\theta(x|z) dz$$

- The posterior is intractable
- Approach: Amortized Variational Inference
- Amortized Variational Inference
  - Choose a good family of approximate posterior distribution $q_\phi()$: Rich, computationally-feasible and efficient $\rightarrow$ Inference networks, model that learns the connection between observations to latent variables
  - Optimize the posterior via efficient computation of the derivatives of the expected loglikelihood $\nabla_\phi E_{q_\phi(z)}[\log p_\theta(x|z)] \rightarrow$ Monte Carlo approximations

Specialty of this problem

- This problem have special properties:
  - Trans-dimensionality: n itself is a variable, which makes the evaluation hard
  - Symmetry: Strong symmetry to the order of objects considered
- Approximate the posterior using a Recurrent Network. This network is run for N steps and will infer at each step the attributes of one object given the image and its previous knowledge of other objects on the image.
- To simplify sequential reasoning, use a unary code z_{pres} for n
- Form of posterior:

$$q_\phi(z, z_{pres}|x) = q_\phi(z^n_{pres} = 0|z^{1:n}, x) \prod_{i=1}^{N} q_\phi(z^i, z^i_{pres} = 1|z^{1:i-1}, x)$$

- Here, z^i_{pres} is a counter. If $z^i_{pres} = 1$, the model need to describe at least one more object. It stops when $z^i_{pres} = 0$.

Learning

- Maximize the negative free energy:

$$L(\theta, \phi) = E_{q_\phi}[\log \frac{p_\theta(x, z, n)}{q_\phi(z, n|x)}]$$

- If p_θ is differentiable, we can estimate $\frac{\partial L}{\partial \theta}$ via Monte Carlo.
- Estimate $\frac{\partial L}{\partial \phi}$ is harder because of n
-
  - Continuous - Gradient Descent: Use the re-parametrization trick in order to back-propagate through zi
  - Discrete - Black box optimization: Use the likelihood ratio estimator.

Experiment 1 - multi-MNIST

- Objective: Learn to detect and generate the constituents digits from scratch.
- An image can include 0,1 or 2 numbers.
- Model:

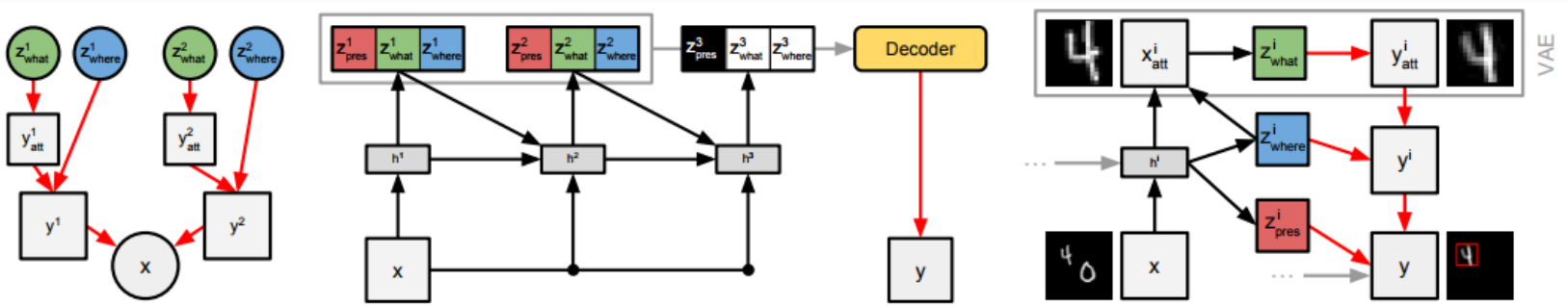


Figure 2: **AIR in practice:** *Left:* The assumed generative model. *Middle:* AIR inference for this model. The contents of the grey box are input to the decoder. *Right:* Interaction between the inference and generation networks at every time-step. In our experiments the relationship between $\mathbf{x}^i_{\text{att}}$ and $\mathbf{y}^i_{\text{att}}$ is modeled by a VAE, however any generative model of patches could be used (even, e.g., DRAW).

Experiment 1 - multi-MNIST

- Train from 60000 such images
- Result:

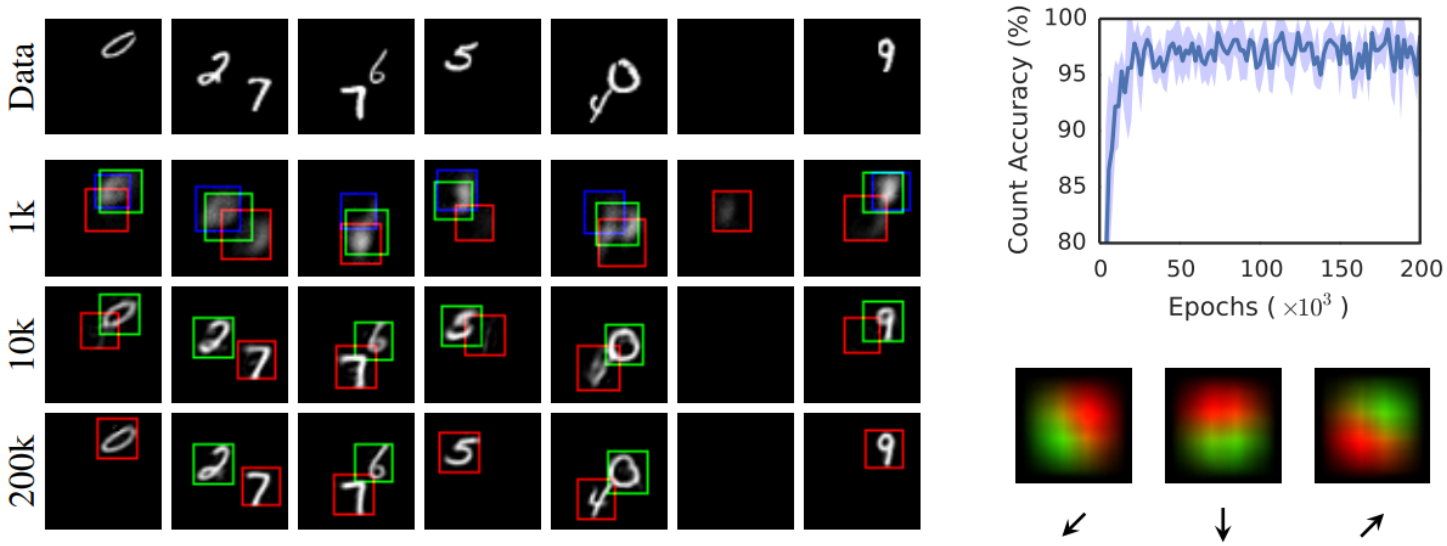


Figure 3: **Multi-MNIST learning:** *Left above:* Images from the dataset. *Left below:* Reconstructions at different stages of training along with a visualization of the model's attention windows. The 1st, 2nd and 3rd time-steps are displayed using red, green and blue borders respectively. A video of this sequence is provided in the supplementary material. *Above right:* Count accuracy over time. The model detects the counts of digits accurately, despite having never been provided supervision. *Below right:* The learned scanning policy for 3 different runs of training (only differing in the random seed). We visualize empirical heatmaps of the attention windows' positions (red, and green for the first and second time-steps respectively). As expected, the policy is random. This suggests that the policy is spatial, as opposed to identity- or size-based.

Generalization

- Train with 0,1,2 digits and test with 3 digits
- Train with 0,1,3 digits and test with 2 digits.

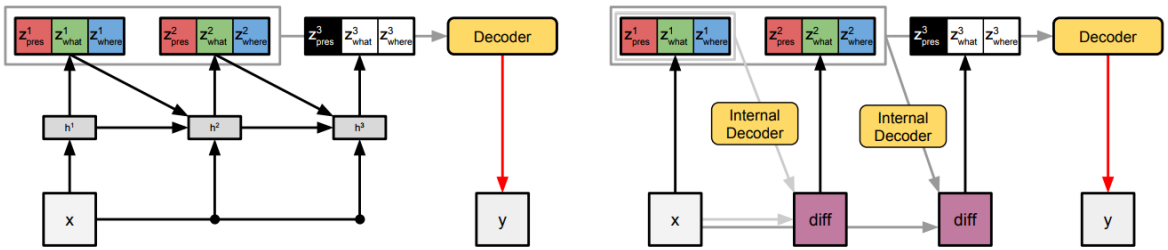


Figure 8: **AIR vs. DAIR:** *Left:* The standard AIR architecture. *Right:* The DAIR architecture. At each time-step i, the latent variables produced so far are used to perform a partial rendering of the scene. The difference of this partial rendering from the image under question is used infer z^i_{pres}, $\mathbf{z}^i_{\text{what}}$ and $\mathbf{z}^i_{\text{where}}$ in the current time-step.

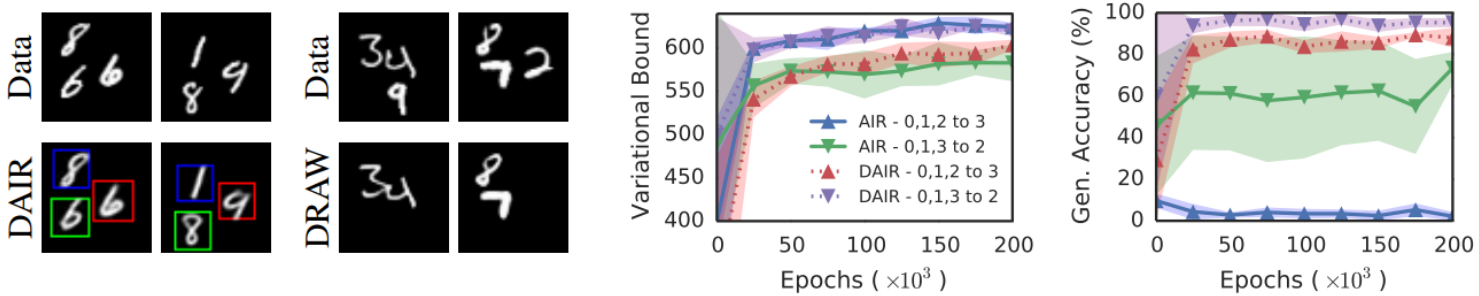


Figure 4: **Strong generalization:** *Left:* Reconstructions of images with 3 digits made by DAIR trained on 0, 1 or 2 digits, as well as a comparison with DRAW. *Right:* Variational lower bound, and generalizing / interpolating count accuracy. DAIR out-performs both DRAW and AIR at this task.

Generate useful represenations

- Train AIR on images containing 0, 1 or 2 digits.
- Then train a second network takes the output of the first one and computes a) the sum of the digits and b) estimates whether they are shown in ascending order.

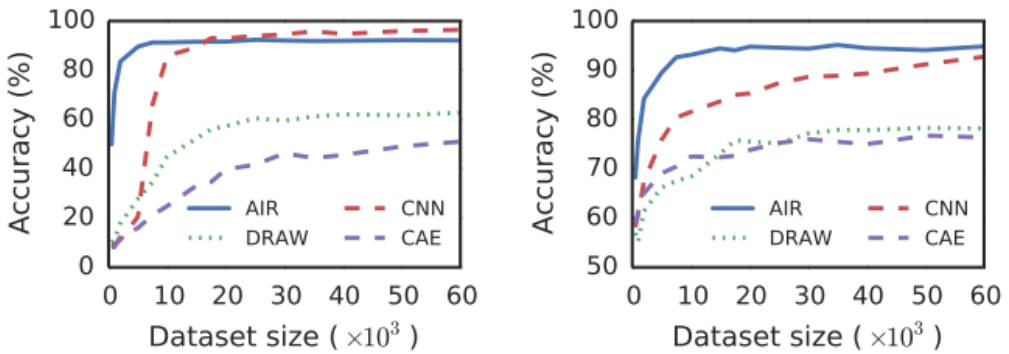


Figure 5: **Representational power:** AIR achieves high accuracy using only a fraction of the labeled data. *Left:* summing two digits. *Right:* detecting if they appear in increasing order. Despite producing comparable reconstructions, CAE and DRAW inferences are less interpretable than AIR's and therefore lead to poorer downstream performance.

Experiment 2 - 3D scene

- Use MuJoCo physics simulator
- Specify a 3D renderer so that it can turned into the 2D image
- To evaluate the likelihood $p(x|z)$, use MuJoGo to render a image y from z and then compute $N(x|y, I\sigma_x^2)$
- Result:

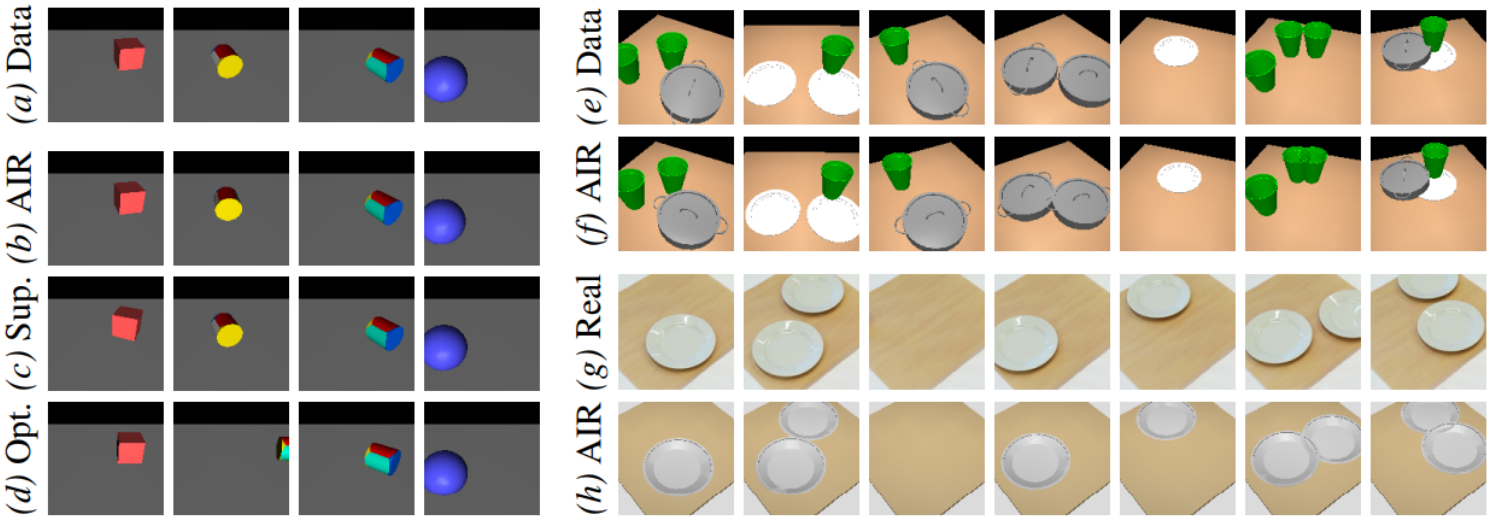


Figure 6: **3D objects:** *Left:* The task is to infer the identity and pose of a single 3D object. (a) Images from the dataset. (b) Unsupervised AIR reconstructions. (c) Supervised reconstructions. Note poor performance on cubes due to their symmetry. (d) Reconstructions after direct gradient descent. This approach is less stable and much more susceptible to local minima. *Right:* AIR can learn to recover the counts, identities and poses of multiple objects in a 3D table-top scene. (e,g) Generated and real images. (f,h) AIR produces fast and accurate inferences which we visualize using the renderer.

Experiment 2 - 3D scene

- Infer the counts, identities and positions of a variable number of crockery items, as well as the camera position, in a table-top scene
- Naive supervised reconstruction suffers when there are repetitions of an item.
- Result:

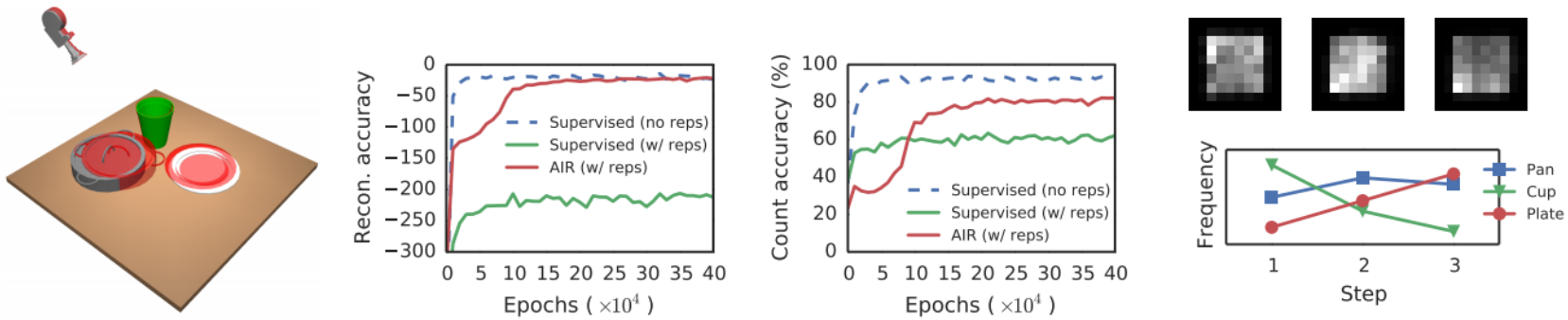


Figure 7: **3D scenes details:** *Left:* Ground-truth object and camera positions with inferred positions overlayed in red (note that inferred cup is closely aligned with ground-truth, thus not clearly visible). We demonstrate fast inference of all relevant scene elements using the AIR framework. *Middle:* AIR produces significantly better reconstructions and count accuracies than a supervised method on data that contains repetitions, and is even competitive on simpler data. *Right:* Heatmap of object locations at each time-step (top). The learned policy appears to be more dependent on identity (bottom).