VAEL: Bridging Variational Autoencoders and Probabilistic Logic Programming

Eleonora Misino, Giuseppe Marra, Emanuele Sansone

Presentation by Lara Vignotto

February 21, 2024

Introduction

- **Neuro-symbolic learning** is emerging as a powerful approach by combining deep learning with logical reasoning, each excelling in different areas.
- Traditional generative models like VAEs are tightly linked to training data labels, limiting their ability to adapt to new tasks without retraining.
- The paper introduces **VAEL**, a neuro-symbolic model that uses a set of logical expressions as a latent representation, allowing for high-level reasoning and generalization beyond the training data.
- VAEL's neural component maps raw data to symbolic representations, reducing dependency on training data and enhancing task generalization.
- The model's reasoning component provides a strong inductive bias, leading to more efficient learning from limited data.

Probabilistic Logic Programming (1)

- **Logic programs** consist of **definite clauses** that act as computational rules, stating that if all conditions (*body literals*) are true, then the conclusion (*head literal*) is true.

$$h \leftarrow b_1 \wedge \ldots \wedge b_n$$

- Definite clauses with no conditions are **facts**.
- **Probabilistic Logic Programs** extend logic programs by introducing **probabilistic facts**, where each fact has an associated probability of being true.

$$p_i :: f_i$$

- **ProbLog** is a system that implements probabilistic logic programs, treating each instance of a probabilistic fact f_i as an *independent Boolean random variable* that is true with probability p_i and false with probability $1 - p_i$.

Probabilistic Logic Programming (2)

- The concept of **possible worlds** is used, where each world is defined by a subset of probabilistic facts and has a probability calculated as the product of the probabilities of the facts being true or false.
- Let $\mathcal{F}$ be the set of all probabilistic facts, and p their corresponding probabilities. Every subset $F \subseteq \mathcal{F}$ defines a **possible world** w_F obtained by adding to F all the atoms that can be derived from F using the logic program. The probability $P\left(w_F; p\right)$ of such a possible world w_F is

$$P\left(w_F; p\right) = \prod_{f_i \in F} p_i \prod_{f_i \in \mathcal{F} \setminus F} \left(1 - p_i\right)$$

Probabilistic Logic Programming (3)

There are two interesting inference tasks on these probabilities:

- **Success probability**: The probability that a query atom or formula is true, calculated as the sum of the probabilities of all worlds where the query is true.

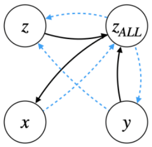
$$P(y; p) = \sum_{F \subseteq \mathcal{F} : w_F \models y} P\left(w_F; p\right)$$

- **Sampling with evidence**: The probability of a world given some evidence, which allows for sampling only from worlds that are consistent with the evidence.

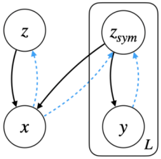
$$P\left(w_F \mid E; p\right) = \frac{1}{Z} \begin{cases} P\left(w_F; p\right) & \text{if } w_F \models E \\ 0 & \text{otherwise} \end{cases}$$

Generation Conditioned on Labels (1)

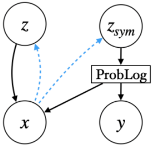
Let us take a look at generative models that use both an image (x) and its label (y) for generation tasks.



(a) M1+M2



(b) CCVAE



(c) VAEL (ours)

Figure 1: Visual comparison for the probabilistic graphical models of [37] (M1+M2), of [31] (CCVAE) and ours (VAEL). Black arrows refer to the generative model, whereas blue dashed arrows correspond to the inference counterpart.

- Model (M1+M2) learns a latent representation that is closely tied to the label, making the image generation dependent on the training task.
- The CCVAE model learns two independent latent vectors, one symbolic (z_{sym}) and one subsymbolic (z), with z_{sym} directly corresponding to the label elements.

Generation Conditioned on Labels (2)

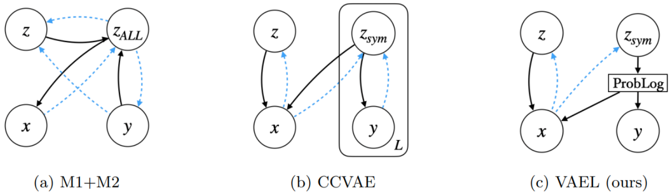
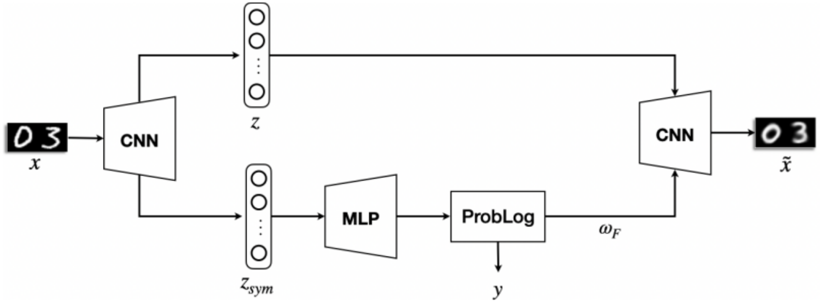


Figure 1: Visual comparison for the probabilistic graphical models of [37] (M1+M2), of [31] (CCVAE) and ours (VAEL). Black arrows refer to the generative model, whereas blue dashed arrows correspond to the inference counterpart.

- These methods are limited because they encode the training task into the latent representation, which is not ideal when the label is only weakly related to the image's symbolic structure.
- How can we generate new images that represent different relationships between digits (like subtraction or multiplication) without retraining on new data, given that existing models are not capable of this flexibility?

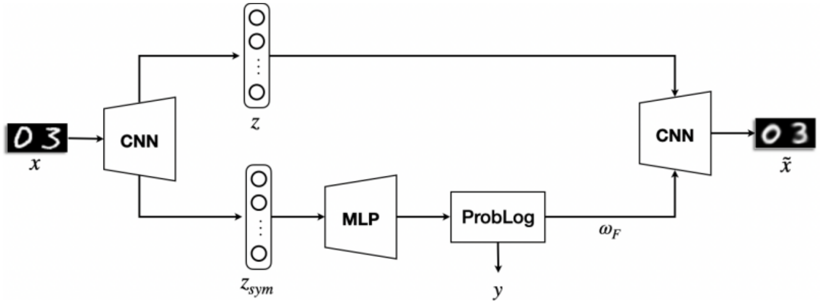
The VAEL Model (1)

The VAEL model integrates three main components: an encoder, a ProbLog program, and a decoder.



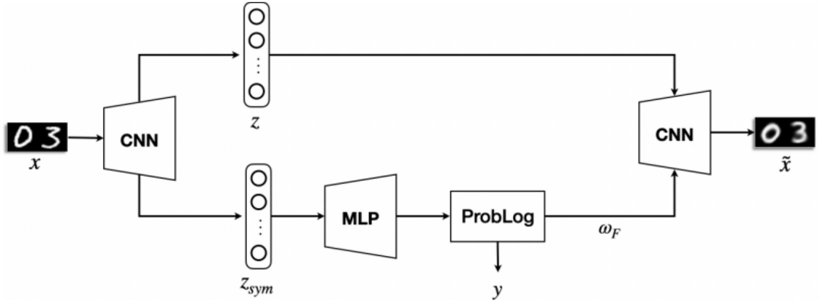
- The **encoder** processes the image x to produce an approximated posterior of latent variables $\mathbf{z}$, which are divided into subsymbolic z and symbolic z_{sym} parts.

The VAEL Model (2)



- The symbolic part, z_{sym}, parameterizes a **ProbLog program** that maps these variables to probabilities, which are then used to compute the label y and a possible world.
- The **decoder** uses both the subsymbolic latent vector z and the possible world from the ProbLog program to reconstruct the image $\tilde{x}$.

The VAEL Model (3)



The four core variables in this model are

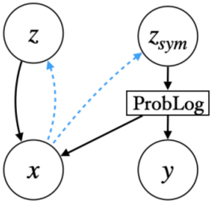
- $x \in \mathbb{R}^{H \times W \times C}$: image we want to generate.
- $y \in \{0,1\}^K$: label (symbolic information characterizing the image).
- $z_{\text{sym}} \in \mathbb{R}^N$: symbolic component of the latent variable.
- $z \in \mathbb{R}^M$: subsymbolic component of the latent variable.

Generative Model

The generative distribution of VAEL is factorized as

$$p_\theta(x, y, \mathbf{z}) = p(x \mid \mathbf{z}) p\left(y \mid z_{\text{sym}}\right) p(\mathbf{z})$$

- $\mathbf{z} = [z_{\text{sym}} \, , z]$
- θ are the parameters of the generative model
- $p(\mathbf{z})$ is a standard Gaussian distribution
- $p\left(y \mid z_{\text{sym}}\right)$ is the success distribution of the label of the ProbLog program T (success probability)
- $p(x \mid \mathbf{z})$ is a Laplace distribution with mean value μ and identity covariance
- μ is a neural network decoder whose inputs are z and ω_F, which is sampled from $P\left(\omega_F; MLP\left(z_{\text{sym}}\right)\right)$ (probability of a world)

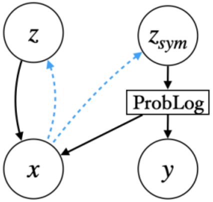


Inference Model

- We amortize inference by using an approximate posterior distribution $q_\phi(\mathbf{z} \mid x, y)$.
- We assume that $\mathbf{z}$ and y are conditionally independent given x

$$q_\phi(\mathbf{z} \mid x, y) = q_\phi(\mathbf{z} \mid x)$$

  using a Gaussian distribution with mean parametrized by the encoder, and identity covariance.
- This allows to decouple the latent representation from the training task (unlike other VAE frameworks).



Objective Function

The objective function of VAEL computes an ELBO on the log likelihood of pair (x, y)

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{REC}(\theta, \phi) + \mathcal{L}_{Q}(\theta, \phi) - \mathcal{D}_{\mathcal{KL}}\left[q_\phi(\mathbf{z} \mid x) \| p(\mathbf{z})\right]$$

Reconstruction error

$$\mathcal{L}_{REC}(\theta, \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid x)}\left[\log(p(x \mid \mathbf{z})\right]$$

Query error

$$\mathcal{L}_{Q}(\theta, \phi) = \mathbb{E}_{z_{\text{sym}} \sim q_\phi(z_{\text{sym}} \mid x))}\left[\log\left(p\left(y \mid z_{\text{sym}}\right)\right)\right]$$

ELBO derivation (1)

Maximization of the log-likelihood of the input x and the class y

$$\log(p(x,y)) = \log\left(\int p(x,y \mid \mathbf{z})d\mathbf{z}\right)$$

Recalling the generative network factorization
$p(x,y,\mathbf{z}) = p(x \mid \mathbf{z})p(y \mid z_{\text{sym}})p(\mathbf{z})$ we can write

$$\log(p(x,y)) = \log\left(\int p_\theta\left(x \mid z, z_{\text{sym}}\right) p_\theta\left(y \mid z_{\text{sym}}\right) p(z) p\left(z_{\text{sym}}\right) dz dz_{\text{sym}}\right)$$

The posterior $p_\theta(\mathbf{z} \mid x)$ is intractable, so we use the approximation $q_\phi(\mathbf{z} \mid x)$

$$\log(p(x,y)) =$$
$$\log\left(\int \frac{q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right)}{q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right)} p_\theta\left(x \mid z, z_{\text{sym}}\right) p_\theta\left(y \mid z_{\text{sym}}\right) p(z) p\left(z_{\text{sym}}\right) dz dz_{\text{sym}}\right)$$

ELBO derivation (2)

$$\log\left(\int \frac{q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right)}{q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right)} p_\theta\left(x \mid z, z_{\text{sym}}\right) p_\theta\left(y \mid z_{\text{sym}}\right) p(z)p\left(z_{\text{sym}}\right) dz dz_{\text{sym}}\right)$$

By Jensen's inequality

$$\int q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right) \log\left(p_\theta\left(x \mid z, z_{\text{sym}}\right) p_\theta\left(y \mid z_{\text{sym}}\right) \frac{p(z)p\left(z_{\text{sym}}\right)}{q_\phi(z \mid x)q_\phi\left(z_{\text{sym}} \mid x\right)} dz dz_{\text{sym}}\right)$$

This is the lower bound for the log-likelihood of x and y, that can be rewrited as

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x)}\left[\log\left(p_\theta(x \mid \mathbf{z})\right)\right] + \mathbb{E}_{z_{\text{sym}}\sim q_\phi(z_{\text{sym}}|x)}\left[\log\left(p_\theta\left(y \mid z_{\text{sym}}\right)\right)\right] + \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x)}\left[\log\left(\frac{p(\mathbf{z})}{q_\phi(\mathbf{z} \mid x)}\right)\right]$$

ELBO derivation (3)

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x)}\left[\log\left(p_\theta(x \mid \mathbf{z})\right)\right] + \mathbb{E}_{z_{\text{sym}}\sim q_\phi(z_{\text{sym}}|x)}\left[\log\left(p_\theta\left(y \mid z_{\text{sym}}\right)\right)\right] + \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x)}\left[\log\left(\frac{p(\mathbf{z})}{q_\phi(\mathbf{z} \mid x)}\right)\right]$$

The last term is the negative KL divergence between the approximate posterior $q_\phi(\mathbf{z} \mid x)$ and the prior $p(\mathbf{z})$. This leads us to the ELBO of the objective function

$$\begin{aligned}
\log(p(x,y)) \geq\ & \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x)}\left[\log\left(p_\theta(x \mid \mathbf{z})\right)\right] && \mathcal{L}_{REC}(\theta,\phi)\\
& + \mathbb{E}_{z_{\text{sym}}\sim q_\phi(z_{\text{sym}}|x)}\left[\log\left(p_\theta\left(y \mid z_{\text{sym}}\right)\right)\right] && \mathcal{L}_{Q}(\theta,\phi)\\
& - \mathcal{D}_{\mathcal{KL}}\left[q_\phi(\mathbf{z} \mid x) \| p(\mathbf{z})\right]\\
:=\ & \mathcal{L}(\theta,\phi)
\end{aligned}$$

ELBO derivation (4)

In VAEL graphical model, ω_F is omitted since the authors exploit an equivalence relation between the probabilistic graphical models (PGMs). The objective for the PGM where ω_F is explicit (right) is equivalent to the one reported in the paper (left).

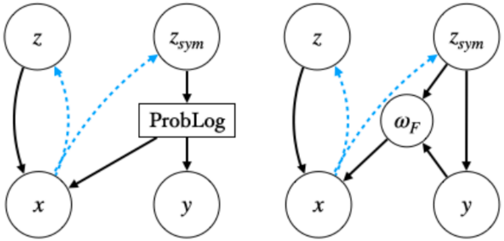


Figure 7: PGM with (left) and without (right) ProbLog box.

Training (1)

- The ELBO and its gradients were estimated w.r.t. the model parameters using standard Monte Carlo estimates of expectations.
- Since both $q_\phi(\mathbf{z} \mid x)$ and $p(\mathbf{z})$ are chosen to be Gaussian distributions, the KL divergence in the objective function can be integrated analytically by relying on its closed form.
- Only the expected errors $\mathcal{L}_{REC}(\theta, \phi)$ and $\mathcal{L}_{Q}(\theta, \phi)$ require estimation by sampling.

ELBO estimator:

$$\mathcal{L}(\theta, \phi) \approx \tilde{\mathcal{L}}(\theta, \phi; \epsilon) = \tilde{\mathcal{L}}_{REC}(\theta, \phi; \epsilon) + \tilde{\mathcal{L}}_{Q}(\theta, \phi; \epsilon) - \mathcal{D}_{\mathcal{KL}}\left[q_\phi(\mathbf{z} \mid x) \| p(\mathbf{z})\right]$$

Training (2)

$$\mathcal{L}(\theta,\phi) \approx \tilde{\mathcal{L}}(\theta,\phi;\epsilon) = \tilde{\mathcal{L}}_{REC}(\theta,\phi;\epsilon) + \tilde{\mathcal{L}}_{Q}(\theta,\phi;\epsilon) - \mathcal{D}_{\mathcal{KL}}\left[q_\phi(\mathbf{z} \mid x) \| p(\mathbf{z})\right]$$

Estimated reconstruction error

$$\tilde{\mathcal{L}}_{REC}(\theta,\phi;\epsilon) = \frac{1}{N}\sum_{n=1}^{N}\left(\log\left(p_\theta\left(x \mid \hat{\mathbf{z}}^{(n)}\right)\right)\right)$$

Estimated query error

$$\tilde{\mathcal{L}}_{Q}(\theta,\phi;\epsilon) = \frac{1}{N}\sum_{n=1}^{N}\left(\log\left(p_\theta\left(y \mid \hat{z}^{(n)}_{\text{sym}}\right)\right)\right)$$

where

$$\hat{\mathbf{z}}^{(n)} = \left\{\hat{z}^{(n)}, \hat{z}^{(n)}_{\text{sym}}\right\} := \mu(x) + \sigma(x)\epsilon^{(n)}, \qquad \epsilon^{(n)} \sim \mathcal{N}(0,1)$$

Training (3)

- During training, the aim is to maximize $\mathcal{L}(\theta, \phi)$ w.r.t both the encoder and the decoder parameters: we need to compute the gradient w.r.t. θ and ϕ.
- Since any sampling operation prevents back-propagation, we need to reparametrize the two sampled variables **z** and ω.
- The authors used for the Gaussian **z** the Reparametrization Trick, and for the discrete variable ω (corresponding to the sampled possible world) they exploited the Categorical Reparametrization with Gumbel-Softmax.

Training (4)

```
Algorithm 1: VAEL Training.
Data: Set of images $\mathcal{X}$
$\theta, \phi \leftarrow$ Initialization of paramters
repeat
    Forward Phase
    $x \leftarrow$ Training sample
    $\mathbf{z} = [z, z_{sym}] \sim q(\mathbf{z} \mid x)$
    $p = MLP(z_{sym})$
    $\omega_F \sim P(\omega_F; p)$
    $y \sim P(y; p)$
    $\tilde{x} \sim p(x|z, \omega_F)$
    Backward Phase
    $\mathbf{g} \leftarrow \nabla_{\theta,\phi}\mathcal{L}(\theta, \phi)$
    $\theta, \phi \leftarrow$ Update parameters using gradients $\mathbf{g}$
until convergence of parameters $(\theta, \phi)$;
```

$$P(w_F; p) = \prod_{f_i \in F} p_i \prod_{f_i \in \mathcal{F} \setminus F} (1 - p_i) \qquad P(y; p) = \sum_{F \subset \mathcal{F} : w_F \models y} P(w_F; p)$$

Experiments

- **Datasets**: A 2digit MNIST dataset with 64,400 images and a Mario dataset with 6,720 images were created for testing.
- **Dataset Splits**: The 2digit MNIST dataset was split into 65% training, 20% validation, and 15% test sets, while the Mario dataset was split into 70% training, 20% validation, and 10% test sets.
- **Evaluation Metrics**: The approach was evaluated using reconstruction loss (mREC), predictive accuracy (mCLASS), and generative accuracy (mGEN).
- **Predictive Accuracy**: For the 2digit MNIST dataset, predictive accuracy is based on the sum of the two digits, and for the Mario dataset, it's based on the agent's move.
- **Generative Accuracy**: An independent classifier was used for each dataset to assess generative accuracy, which involves generating an image and label, splitting the image, classifying the parts, and comparing the results with the generated label.
- **Model Comparison**: VAEL was compared with CCVAE where possible.

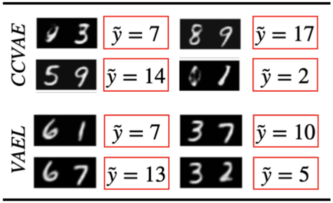
Label Classification

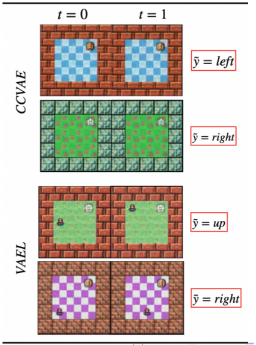
- **Goal**: Predict the correct label from the input image, evaluated by predictive accuracy (mCLASS).
- **Method**: VAEL uses ProbLog inference, while CCVAE uses a neural network to parameterize ($p(y|z_{\text{sym}})$).

Dataset	**Model**	$m_{REC}(\downarrow)$	$m_{CLASS}(\uparrow)$	$m_{GEN}(\uparrow)$
2digit MNIST	*CCVAE*	1549 ± 2	0.5284 ± 0.0051	0.5143 ± 0.0157
	VAEL	$\mathbf{1542 \pm 3}$	$\mathbf{0.8477 \pm 0.0178}$	$\mathbf{0.7922 \pm 0.0350}$
Mario	*CCVAE*	43461 ± 209	$\mathbf{1.0 \pm 0.0}$	0.0 ± 0.0
	VAEL	$\mathbf{42734 \pm 246}$	$\mathbf{0.977 \pm 0.0585}$	$\mathbf{0.8135 \pm 0.2979}$

Image Generation

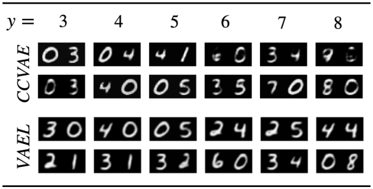
- **Goal**: Assess model performance in generating both the image and its label.
- **Method**: VAEL generates from a sampled latent vector ($\mathbf{z} \sim \mathcal{N}(0,1)$), while CCVAE samples a label first, then the latent vector, and finally the image.

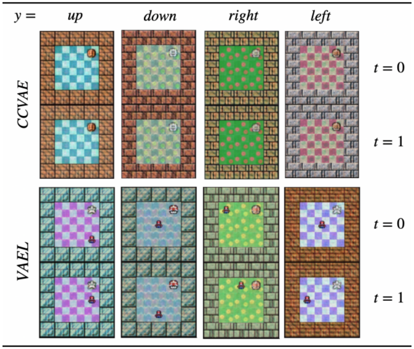




Conditional Image Generation

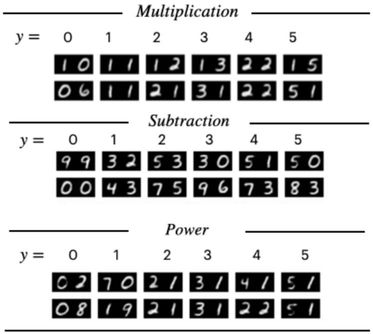
- **Goal**: Evaluate the model's ability to generate images conditionally based on evidence.
- **Method**: VAEL consistently generates coherent pairs of digits and states, while CCVAE often fails to meet the evidence criteria.

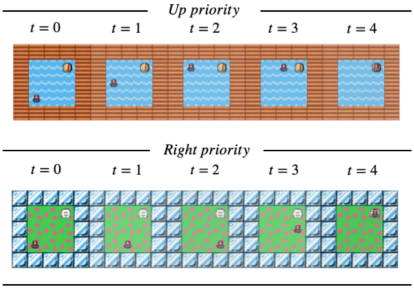




Task Generalization

- **Goal**: Test VAEL's ability to generalize to new tasks without retraining.
- **Approach**: For the 2digit MNIST dataset, tasks like multiplication, subtraction, and exponentiation were introduced. For the Mario dataset, two shortest path tasks were defined.




Up priority
$t = 0$ $t = 1$ $t = 2$ $t = 3$ $t = 4$
Right priority
$t = 0$ $t = 1$ $t = 2$ $t = 3$ $t = 4$

Controlled Image Generation

- Generative models are categorized into those based on **text descriptions** and those using **scene graphs**.
- **Text-based models** focus on controlling *object properties* (shape, color, texture), *spatial relations* (positioning of objects relative to each other), or *both properties and relations*.
- Unlike previous works, VAEL framework employs probabilistic logic programming to encode and reason with first-order logical knowledge.
- This method allows for generalization to both compositions of known relations and entirely new relations.
- **Scene graph-based models** explicitly encode object relations but are less expressive than the logical programs used in this framework, which enable more general reasoning capabilities.

Unsupervised Scene Decomposition

Unsupervised scene decomposition is categorized into:

- **Object-oriented approaches**: Focus on learning representations of individual objects to reconstruct images or sequences.
- **Part-oriented approaches**: Decompose objects into their basic parts for reconstruction.
- **Hierarchical approaches**: Combine the above two to decompose scenes into objects and their parts.

Current Methods use scene-mixtures, spatial attention models, and combinations for *object-oriented decomposition*, and employ encoders and decoders for *part-oriented decomposition*. Recent efforts aim to achieve *hierarchical decomposition*, learning both objects and parts.

This paper's approach focuses on static images, not using temporal information. Does not rely on predefined information about object locations. Utilizes a simple autoencoder architecture to discover objects based on logical relations.

Neuro-Symbolic Generation

- Neuro-symbolic generation is a new field combining machine learning with logical reasoning.
- Previous models had a two-layered latent representation for scenes and objects but were limited to specific spatial relations.
- VAEL model uses a **logical reasoning framework** to handle a wider range of generative tasks and knowledge manipulation.
- Attempts to integrate generative models with probabilistic programming have been made, but they often had limited reasoning capabilities.
- This paper's approach provides a **unified model** that can generate images and perform logical reasoning simultaneously.
- The authors are the first to conduct experiments showing the **benefits of this integration** in terms of task generalization and data efficiency.

Neuro-Symbolic Generation

- Neuro-symbolic generation is a new field combining machine learning with logical reasoning.
- Previous models had a two-layered latent representation for scenes and objects but were limited to specific spatial relations.
- VAEL model uses a **logical reasoning framework** to handle a wider range of generative tasks and knowledge manipulation.
- Attempts to integrate generative models with probabilistic programming have been made, but they often had limited reasoning capabilities.
- This paper's approach provides a **unified model** that can generate images and perform logical reasoning simultaneously.
- The authors are the first to conduct experiments showing the **benefits of this integration** in terms of task generalization and data efficiency.

Conclusions

- **VAEL Model**: Introduced as a neuro-symbolic generative model combining Variational Autoencoders (VAE) with Probabilistic Logic Programming.
- **Symbolic Component**: Enables the model to separate its internal understanding from specific tasks, leading to powerful generalization capabilities.
- **Performance**: Demonstrated state-of-the-art image generation abilities in benchmarks, excelling even with limited data and across various prediction tasks.
- **Future Improvements**: Plans to explore more scalable options for probabilistic programs, such as stochastic logic programs.
- **Broader Applications**: Aims to apply VAEL in different domains like structured object generation to further demonstrate its flexibility and expressiveness.

Thank you!

All material and images can be found in the original paper
`https://papers.nips.cc/paper_files/paper/2022/hash/1e38b2a0b77541b14a3315c99697b835-Abstract-Conference.html`