Probabilistic Neural Programmed Networks for Scene Generation

Zhiwei Deng, Jiacheng Chen, Yifang Fu, Greg Mori

Simon Fraser University

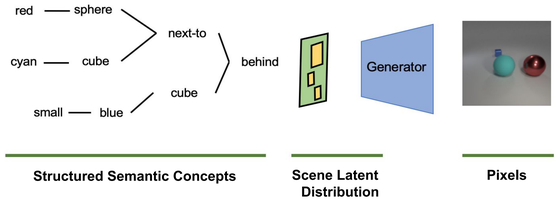
32nd Conference on
Neural Information Processing Systems
(NeurIPS 2018)

Overview

Generating scenes from rich and complex semantics is an important step towards understanding the visual world. To generate scenes from complex semantic descriptions, we develop a flexible latent distribution built from reusable, modular components

Our Method: Probabilistic Neural Programmed Networks (PNP-Net)

- Programmatic / Modular
- Probabilistic
- Compositional
- Reusable



Problem Setting: rich semantics to complex scenes

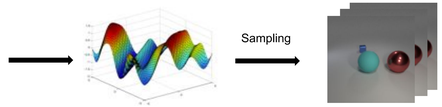
Semantics with different forms:
- sentence
- syntax tree
- scene graph
- ...

Complex scenes that conform to input semantics:
- deterministic images
- a distribution of images constrained by the semantics

Typical semantics contain:

Attributes, Objects, Relations

A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.



Challenges: Complex scenes contain varied **visual elements**, compositional **visual concepts**, and complicated **relations** between objects, which makes the generation highly challenging.

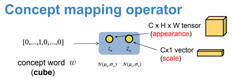
- An ideal generative model should encompass three core components:
  - An interpretable generation process that composes complex scenes
  - What are the primitive visual elements and how are they composed
  - The rendering of abstract concepts into their pixel-level realizations
- Monolithic generative models can generate images of single objects well, but have difficulties handling more complex scenes

Contributions

We propose **Probabilistic Neural Programmed Networks (PNP-Net)**, a programmatic generative model framework for complex scene generation. Specifically, we:
- Introduce a set of **visual elements** and **neural modules/programs** for modifying the appearance of these visual elements
- Integrate our probabilistic neural modules into the canonical VAE framework and generalize the VAE by empowering it with **reusable, composable, interpretable modules**
- Demonstrate **generalization ability** for complicated scene understanding, including zero-shot learning of novel semantic compositions

PNP-Net: Probabilistic operators

Concept mapping operator

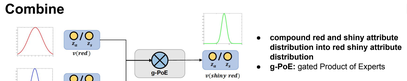


$$\tau_{concept}(w) = [p(\boldsymbol{z}_{\boldsymbol{a}}; \boldsymbol{f}^{\boldsymbol{a}}_{\boldsymbol{\mu}}(w), \boldsymbol{f}^{\boldsymbol{a}}_{\boldsymbol{\sigma}}(w)), p(\boldsymbol{z}_{\boldsymbol{s}}; \boldsymbol{f}^{\boldsymbol{s}}_{\boldsymbol{\mu}}(w), \boldsymbol{f}^{\boldsymbol{s}}_{\boldsymbol{\sigma}}(w))]$$

Note: All distributions in PNP-Net's framework are multivariate Gaussian, they are represented represented by mean and variance in the whole compositional procedure

Aggregation operator

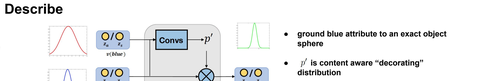
Combine



- **compound red and shiny attribute distribution into red shiny attribute distribution**
- **g-PoE**: gated Product of Experts

$$\tau_{combine}(p_i, p_j) = \frac{1}{Z} p(z_i; g_i'' \circ \mu_i, g_i'' \circ \sigma_i) p(z_j; g_i'' \circ \mu_j, g_i'' \circ \sigma_j)$$

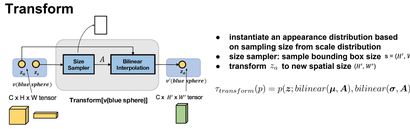
Describe



- **ground blue attribute to an exact object sphere**
- **p' is content aware "decorating" distribution**

$$\tau_{describe}(p_i, p_j) = p'(z'; f_\mu(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j), f_\sigma(\sigma_i, \sigma_j)) \otimes p_j(z_j; \mu_j, \sigma_j)$$

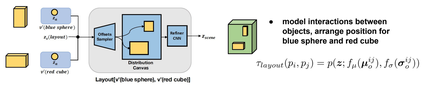
Transform



- **instantiate an appearance distribution based on sampling size from scale distribution**
- **size sampler: sample bounding box size** $s = (h', W')$
- **transform** z_0 **to new spatial size** (h', W')

$$\tau_{transform}(p) = p(\boldsymbol{z}; bilinear(\boldsymbol{\mu}, \boldsymbol{A}), bilinear(\boldsymbol{\sigma}, \boldsymbol{A}))$$

Layout



- **model interactions between objects, arrange position for blue sphere and red cube**

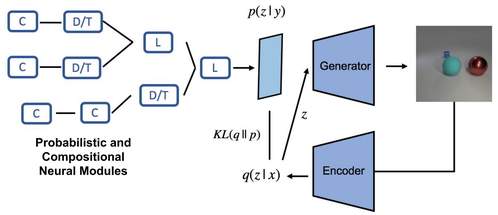
$$\tau_{layout}(p_i, p_j) = p(\boldsymbol{z}; f_\mu(\boldsymbol{\mu}_i^{(j)}), f_\sigma(\boldsymbol{\sigma}_i^{(j)}))$$

References
1. S. Reed et al. Neural programmer-interpreters. In ICLR, 2016.
2. J. Andreas et al. Deep compositional question answering with neural module networks. In CVPR, 2016.
3. E. Parisotto et al. Neuro-symbolic program synthesis.
4. T. Anh Le, et al. Inference compilation and universal probabilistic programming. In AISTATS 2017.

PNP-Net: Overall Formulation

We integrate our probabilistic neural modules into canonical VAE framework, the reusable neural modules **compose the prior of images constrained by given semantics**.

$$\begin{aligned}\log p(\boldsymbol{x}|\boldsymbol{y}) - KL(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{y}) - \log q(\boldsymbol{z}|\boldsymbol{x})] \\ &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - KL(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{y})) \\ &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - KL(q(\boldsymbol{z}|\boldsymbol{x})||\mathcal{P}(\tau, \boldsymbol{y}))\end{aligned}$$



Learning framework of PNP-Net

Notations

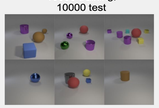
$\boldsymbol{x}$	*scene image*
$\boldsymbol{y}$	*semantics*
$\boldsymbol{z}$	*scene latent*
$p(\boldsymbol{x}\mid\boldsymbol{y})$	*data likelihood*
$p(\boldsymbol{z}\mid\boldsymbol{y},\boldsymbol{x})$	*true posterior distribution*
$q(\boldsymbol{z}\mid\boldsymbol{x})$	*variational distribution*
$p(\boldsymbol{z}\mid\boldsymbol{y})$	*learnable prior*
$\mathcal{P}(\tau,\boldsymbol{y})$	*compositional generation process*

Datasets

Color MNIST
8000 training, 8000 test

- an variant of MNIST
- an image contains at most 2 digits on different locations
- digit has different color, size and style
- comparisons with monolithic generative models

CLEVR-G
10000 training, 10000 test



- an variant of CLEVR designed for image generation task
- up to 8 objects
- different shapes, colors, materials, relations
- comparisons with monolithic generative models
- extra experiments for generalizability

Experiments

Performance Measure

Detector score for measuring semantic correctness: **mAP** of Faster-RCNN object detector

- **Objectness**: whether the number of objects are consistent with semantics?
- **Object type**: whether the objects in generated images have correct type (shape)?
- **Object-attribute**: whether the objects in generated images have correct visual attributes?

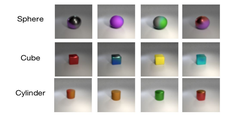
Color-MNIST

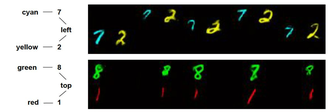
Method	OBJ-N	OBJ-T	OBJ-A
GT	0.999	0.999	0.999
DC-GAN	**0.990**	0.211	0.146
PixelCNN	0.921	**0.474**	0.318
PoE-VAE	0.913	0.136	0.051
Ours	0.981	0.419	**0.363**

CLEVR-G(64x64)

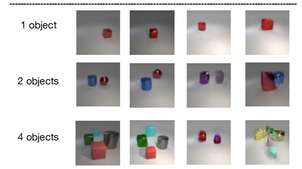
Method	OBJ-N	OBJ-T	OBJ-A
GT	0.999	0.998	0.976
DC-GAN	**0.979**	0.566	0.176
PixelCNN	0.894	0.444	0.074
PoE-VAE	0.974	0.493	0.134
Ours	0.971	**0.833**	**0.737**

Illustration of Visual Concepts





Qualitative Comparisons



Experiments: Generalizability

CLEVR-G(64x64): Zero-shot setting

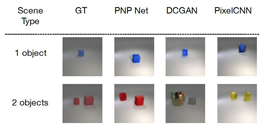
The test set contains combinations of objects, attributes and relations that are not present in training set

Method	OBJ-N	OBJ-T	OBJ-A
DC-GAN	**0.989**	0.418	0.332
PixelCNN	0.899	0.420	0.192
PoE-VAE	0.975	0.441	0.318
Ours	0.970	**0.752**	**0.734**

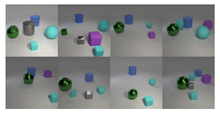
CLEVR-G(128x128): More complex scenes

We can train the model on simpler scenes, but test it on more complex cases

Settings(trained on)	OBJ-N	OBJ-T	OBJ-A
GT(-)	0.979	0.977	0.943
Ours (up-to-8)	0.973	0.734	0.567
Ours (up-to-4)	0.976	0.715	0.518



Generated Complex Scenes



More Details

Please check our project page on Github for more details about model implementation and data.