Deep Generative Models with Learnable Knowledge Constraints

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Xiaodan Liang, Lianhui Qin, Haoye Dong, Eric P. Xing
Carnegie Mellon University   Petuum Inc.

Overview

- Rich deep generative models (DGMs): GANs, VAEs, auto-regessive nets
- Difficult to exploit problem structures and domain knowledge (e.g., human body structure in image generation, Fig.1) in these DGMs.

Existing approaches:

- A popular way of adding structured knowledge with deep neural networks is to design *specialized neural architectures*
  - E.g., Conv-pooling architecture of ConvNet to hard-code translation-invariance of image classification
  - Usually only applicable to specific knowledge, models, or tasks
- *Posterior Regularization* (**PR**) is a principled framework to impose knowledge constraints on posterior distributions of probabilistic models [1] or neural networks [2]. But with difficulties:
  - Many of the DGMs are *not* formulated with the probabilistic Bayesian framework and do not possess a posterior distribution or even meaningful latent variables
  - Require *a priori* fixed constraints. Users have to fully specify the constraints beforehand — impractical due to heavy engineering; sub-optimal without adaptivity to the data and models.

This paper:

- A general means of incorporating **arbitrary structured knowledge** with **any types of deep (generative) models** in a principled way.
- Formal connections between PR and reinforcement learning (**RL**)
- Extends PR to **learn constraints** as the extrinsic reward in RL

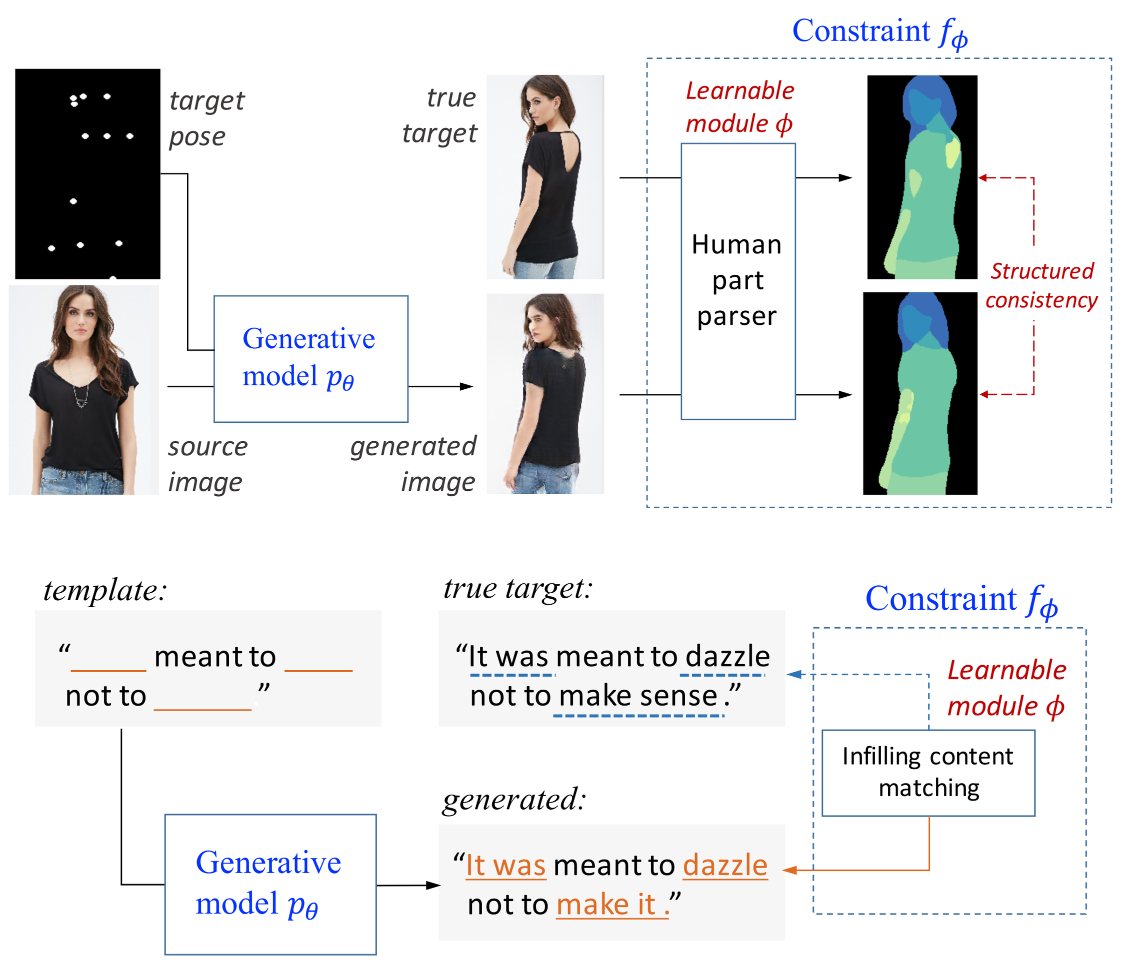

Figure 1: Two examples of imposing learnable knowledge constraints on DGMs. **(a)** Given a person image and a target pose (defined by key points), the goal is to generate an image of the person under the new pose. The constraint is to force the human parts (e.g., head) of the generated image to match those of the true target image. **(b)** Given a text template, the goal is to generate a complete sentence following the template. The constraint is to force the match between the infilling content of the generated sentence with the true content.

Connecting Posterior Regularization (PR) to RL

1) (Adapted) PR for Deep Generative Models (DGMs)

- Consider a generative model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$ with parameters $\boldsymbol{\theta}$
- Consider constraint function $f(\boldsymbol{x}) \in \mathbb{R}$. A higher $f(\boldsymbol{x})$ value indicates a better $\boldsymbol{x}$ in terms of the particular knowledge.
- PR assumes a variational distribution q, and the objective:

$$\min_{\theta,q} \mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})) - \alpha \mathbb{E}_q\left[f(\boldsymbol{x})\right], \tag{1}$$

which is solved with an EM-style procedure

$$\begin{aligned} &\text{E-step:} \quad q^*(\boldsymbol{x}) = p_\theta(\boldsymbol{x}) \exp\left\{\alpha f(\boldsymbol{x})\right\}/Z, \\ &\text{M-step:} \quad \min_\theta \mathrm{KL}(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})) = \min_\theta -\mathbb{E}_q\left[\log p_\theta(\boldsymbol{x})\right] + const. \end{aligned} \tag{2}$$

- In PR, constraint f is fixed. It's sometimes desirable or necessary to enable learnable constraints so that practitioners are allowed to specify only the known components of f while leaving any unknown or uncertain components automatically learned (e.g., the human part parser in Fig.1).
- Denote the constraint function with learnable components as $f_\phi(\boldsymbol{x})$

2) Entropy-Regularized Policy Optimization (ERPO)

- ERPO augments policy gradient with information theoretic regularizers e.g., KL divergence between new and old policies for stabilized learning.
- Assume state s, action a, policy $p_\pi(a|s)$, reward $R(s,a) \in \mathbb{R}$
- Let $\boldsymbol{x} = (s,a)$ denote the state-action pair, and $p_\pi(\boldsymbol{x}) = \mu^\pi(s)p_\pi(a|s)$ where $\mu^\pi(s)$ is the stationary state distribution.
- Let $q_\pi(x)$ be the new policy; $p_\pi(x)$ the old. In some ERPO such as *relative entropy policy search*, q_π is non-parametric. Objective:

$$\min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x})\|p_\pi(\boldsymbol{x})) - \alpha \mathbb{E}_{q_\pi}\left[R(\boldsymbol{x})\right], \tag{3}$$

Close resemblance between Eq.(1) and Eq.(3):

- Generative model $p_\theta(\boldsymbol{x})$ in PR $\Leftrightarrow$ reference (old) policy $p_\pi(\boldsymbol{x})$
- Constraint f in PR $\Leftrightarrow$ reward R
- Solution for q_π is in the same form of Eq.(2)

3) Maximum-Entropy Inverse Reinforcement Learning (MaxEnt IRL)

- Learns reward $R_\phi(\boldsymbol{x})$ with unknown parameters $\boldsymbol{\phi}$.
- Assumes p_π a uniform $\rightarrow q_\phi(\boldsymbol{x}) := \exp\{\alpha R_\phi(\boldsymbol{x})\}/Z_\phi$. Learns $\boldsymbol{\phi}$ with:

$$\boldsymbol{\phi}^* = \arg\max_\phi \mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\log q_\phi(\boldsymbol{x})\right]. \tag{4}$$

Components	PR	Entropy-Reg RL	MaxEnt IRL
$\boldsymbol{x}$	data/generations	action-state samples	demonstrations
$p(\boldsymbol{x})$	generative model p_θ	(old) policy p_π	—
$f(\boldsymbol{x})/R(\boldsymbol{x})$	constraint f_ϕ	reward R	reward R_ϕ
$q(\boldsymbol{x})$	variational distr. Eq.2	(new) policy q_π	policy q_ϕ

Table 1: Mathematical correspondence of PR with the entropy-regularized RL and maximum entropy IRL.

Algorithm

With the connection between PR and RL, we can transfer the MaxEnt IRL technique of reward learning for constraint learning. The resulting algorithm alternates the optimization of constraint f_ϕ and model p_θ.

Learning the Constraint f_ϕ

Use the same objective of MaxEnt IRL (Eq.1), replacing q_ϕ with $q(\boldsymbol{x})$ from Eq.2:

$$\begin{aligned} \nabla_\phi \mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\log q(\boldsymbol{x})\right] &= \nabla_\phi\left[\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\alpha f_\phi(\boldsymbol{x})\right] - \log Z_\phi\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\alpha \nabla_\phi f_\phi(\boldsymbol{x})\right] - \mathbb{E}_{q(\boldsymbol{x})}\left[\alpha \nabla_\phi f_\phi(\boldsymbol{x})\right]. \end{aligned} \tag{5}$$

Learning the Generative Model p_θ

Given the current parameter state $(\boldsymbol{\theta} = \boldsymbol{\theta}^t, \boldsymbol{\phi} = \boldsymbol{\phi}^t)$, and $q(\boldsymbol{x})$ evaluated at the parameters, we continue to update the generative model.

- For *explicit model*, we use the M-step as in Eq.(2):

$$\min_\theta \mathrm{KL}(q(\boldsymbol{x})\|p_\theta(\boldsymbol{x})) = \min_\theta -\mathbb{E}_{q(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right] + const. \tag{6}$$

- For *implicit model* that permits only simulating samples but not evaluating density, we propose to minimize the *reverse* KL divergence:

$$\min_\theta \mathrm{KL}\left(p_\theta(\boldsymbol{x})\|q(\boldsymbol{x})\right) = \min_\theta -\mathbb{E}_{p_\theta}\left[\alpha f_{\phi^t}(\boldsymbol{x})\right] + \mathrm{KL}(p_\theta\|p_{\theta^t}) + const. \tag{7}$$

See paper for efficient approximations and connections to GANs.

Experiments

Method	SSIM	Human
Energy-based GAN	0.716	–
Base model	0.676	0.03
W/ fixed constraint	0.679	0.12
W/ learned constraint	**0.727**	**0.77**

Model	PPL	Human
Base model	30.30	0.19
W/ binary D	30.01	0.20
W/ learned constraint	**28.69**	**0.24**

Table 2: Results of Human Pose Image Generation (Left, Fig.2(a)) and Template Guided Sentence Generation (Right, Fig.2(b)). Pls see the paper for more details.

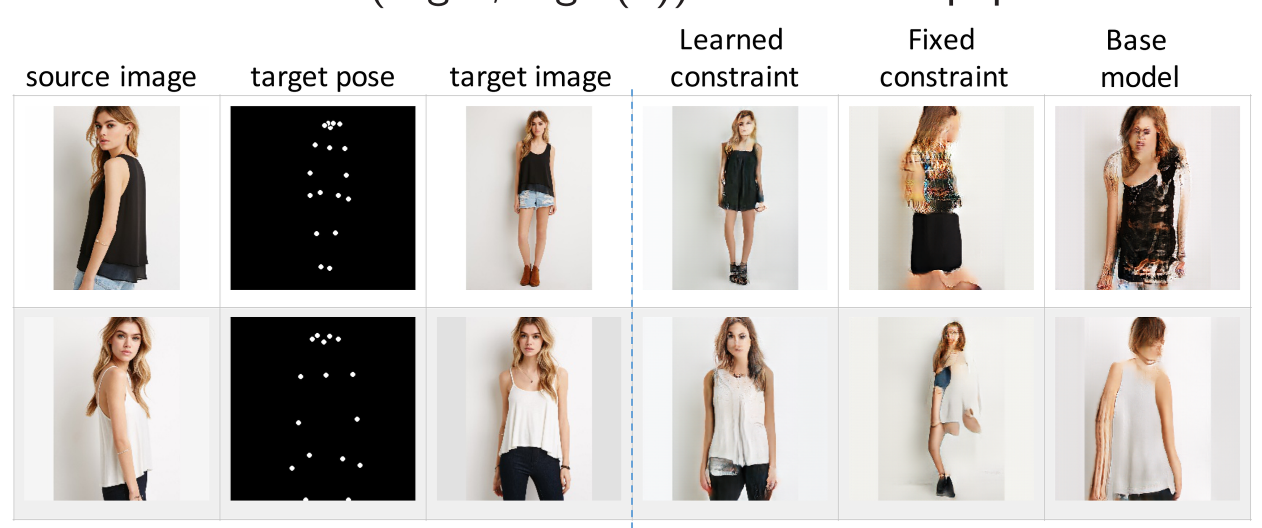

Figure 2: Generation samples.

References

[1] K. Ganchev, J. Gillenwater, B. Taskar, et al. "Posterior regularization for structured latent variable models". In: *JMLR* 11.Jul (2010), pp. 2001–2049.
[2] Z. Hu et al. "Harnessing deep neural networks with logic rules". In: *ACL*. 2016.