Deep Generative Models with Learnable Knowledge Constraints

Zhiting Hu[1,2], Zichao Yang[1], Ruslan Salakhutdinov[1],

Xiaodan Liang[1], Lianhui Qin, Haoye Dong, Eric P. Xing[1,2]

Carnegie Mellon University[1]

Petuum Inc[2]

Recent advances in deep generative models

- Deep generative models (DGMs)
  - Variational autoencoders (VAEs) [Kingma & Welling, 2013]
  - Generative adversarial networks (GANs) [Goodfellow et al., 2014]
  - Auto-regressive models
- Impressive success in vision/text domains

Recent advances in deep generative models

- Deep generative models (DGMs)
  - Variational autoencoders (VAEs) [Kingma & Welling, 2013]
  - Generative adversarial networks (GANs) [Goodfellow et al., 2014]
  - Auto-regressive models
- Impressive success in vision/text domains

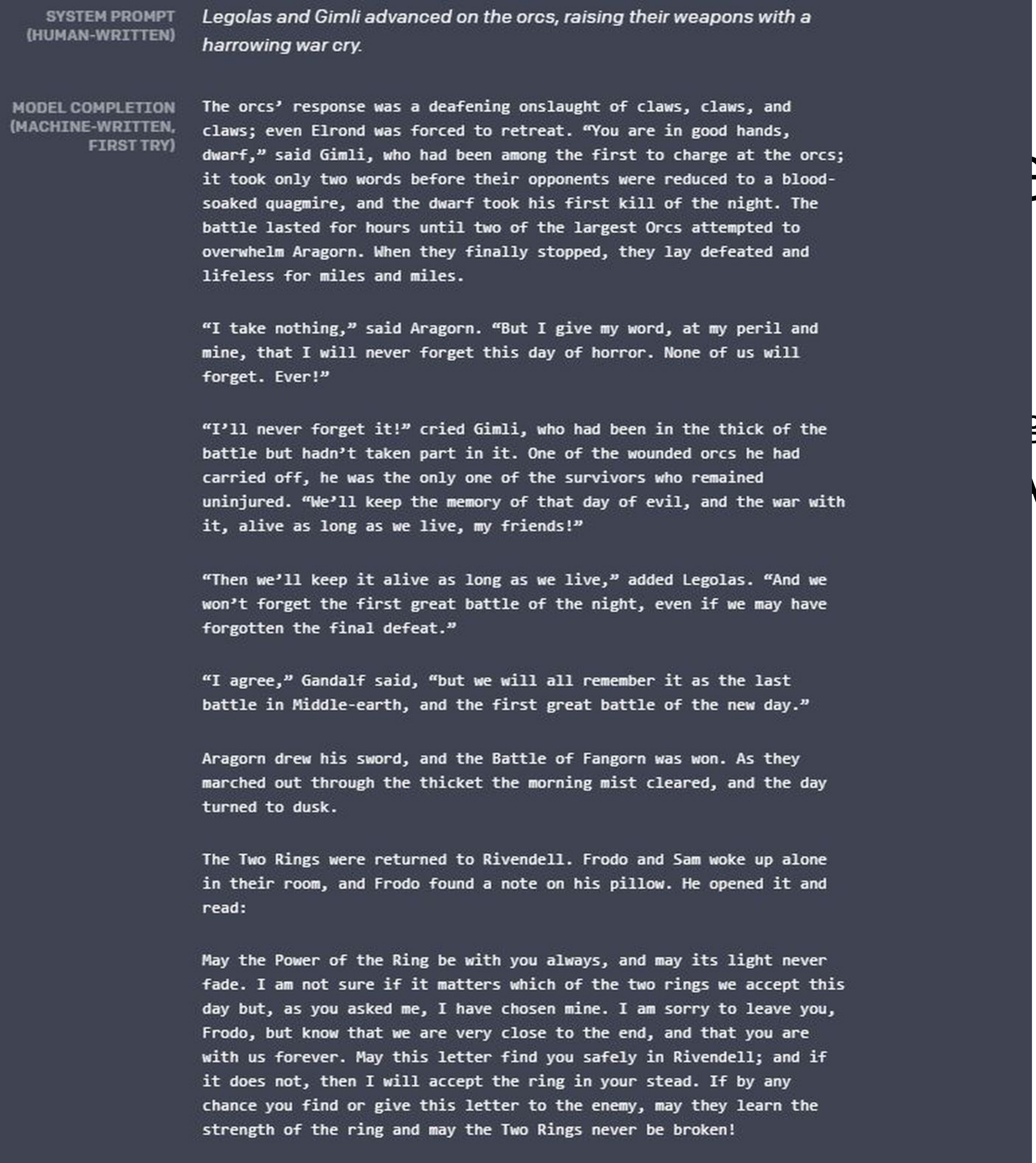
GAN-generated face images by NVIDA [Karras et al., 2018]

ep generative models

gma & Welling, 2013]

ANs) [Goodfellow et al., 2014]

omains



SYSTEM PROMPT (HUMAN-WRITTEN)

Legolas and Gimli advanced on the orcs, raising their weapons with a harrowing war cry.

MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)

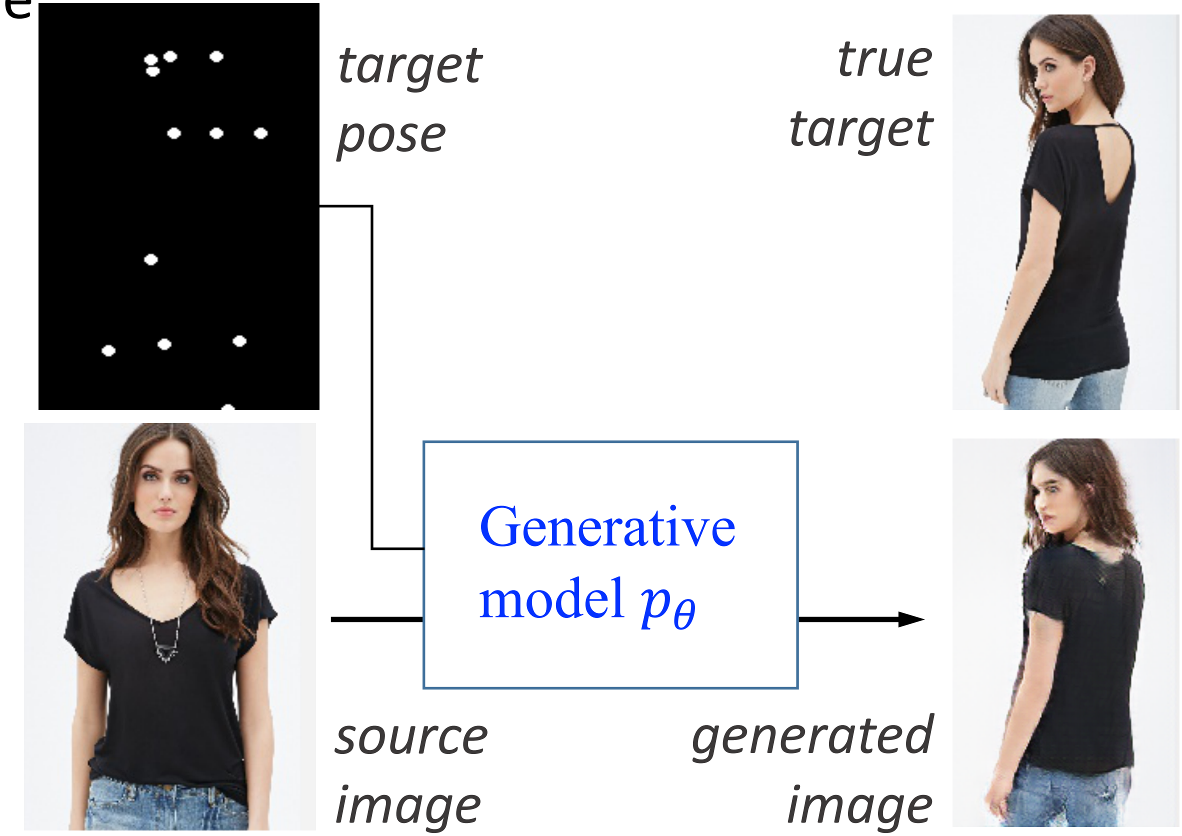
```
The orcs' response was a deafening onslaught of claws, claws, and
claws; even Elrond was forced to retreat. "You are in good hands,
dwarf," said Gimli, who had been among the first to charge at the orcs;
it took only two words before their opponents were reduced to a blood-
soaked quagmire, and the dwarf took his first kill of the night. The
battle lasted for hours until two of the largest Orcs attempted to
overwhelm Aragorn. When they finally stopped, they lay defeated and
lifeless for miles and miles.

"I take nothing," said Aragorn. "But I give my word, at my peril and
mine, that I will never forget this day of horror. None of us will
forget. Ever!"

"I'll never forget it!" cried Gimli, who had been in the thick of the
battle but hadn't taken part in it. One of the wounded orcs he had
carried off, he was the only one of the survivors who remained
uninjured. "We'll keep the memory of that day of evil, and the war with
it, alive as long as we live, my friends!"

"Then we'll keep it alive as long as we live," added Legolas. "And we
won't forget the first great battle of the night, even if we may have
forgotten the final defeat."

"I agree," Gandalf said, "but we will all remember it as the last
battle in Middle-earth, and the first great battle of the new day."

Aragorn drew his sword, and the Battle of Fangorn was won. As they
marched out through the thicket the morning mist cleared, and the day
turned to dusk.

The Two Rings were returned to Rivendell. Frodo and Sam woke up alone
in their room, and Frodo found a note on his pillow. He opened it and
read:

May the Power of the Ring be with you always, and may its light never
fade. I am not sure if it matters which of the two rings we accept this
day but, as you asked me, I have chosen mine. I am sorry to leave you,
Frodo, but know that we are very close to the end, and that you are
with us forever. May this letter find you safely in Rivendell; and if
it does not, then I will accept the ring in your stead. If by any
chance you find or give this letter to the enemy, may they learn the
strength of the ring and may the Two Rings never be broken!
```

Machine-written text by OpenAI [Radford et al., 2019]

GAN-generated face images by NVIDA [Karras et al., 2018]

Difficulty in exploit problem structures and domain knowledge

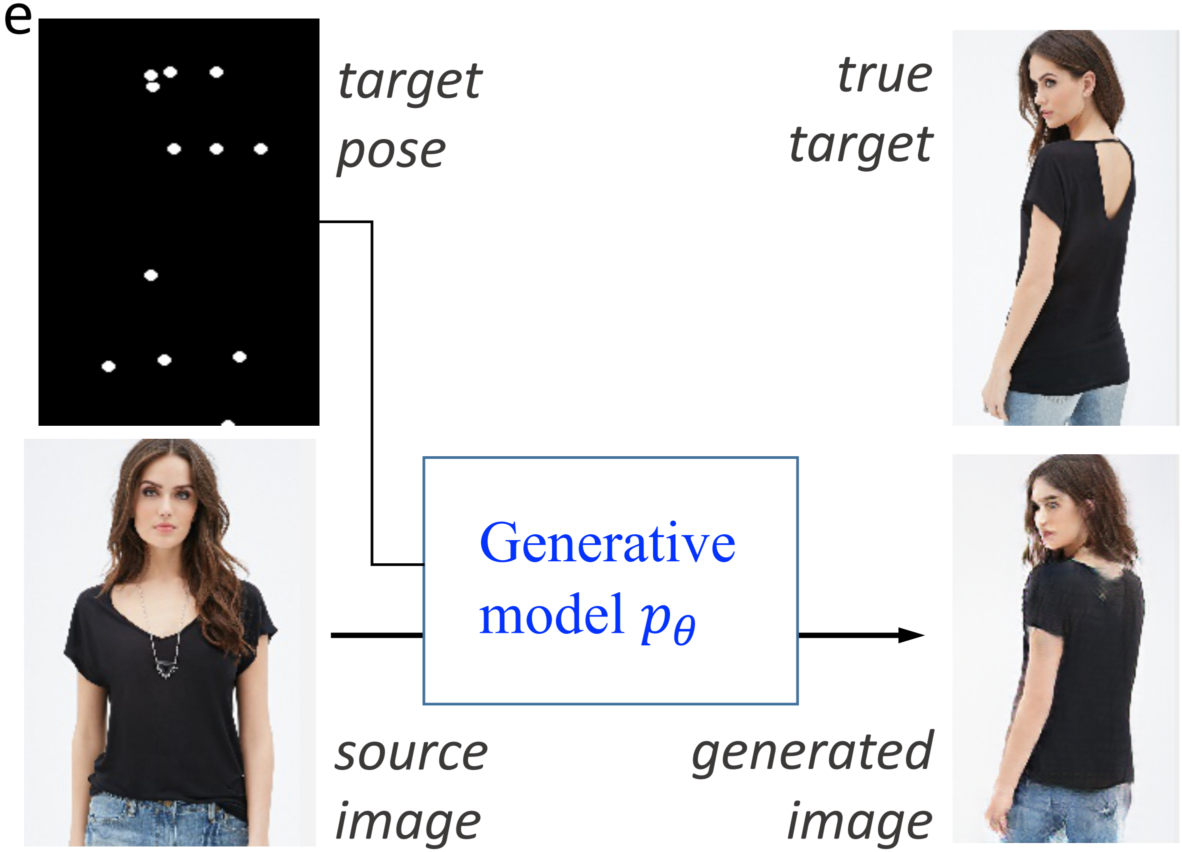
- Pose Conditional Person Image Generation
  - Given a person image and a target pose, generate an image of the person under the new pose



Difficulty in exploit problem structures and domain knowledge

- Pose Conditional Person Image Generation
  - Given a person image and a target pose, generate an image of the person under the new pose

- Generative models can be trained on **supervised data**



target pose

true target

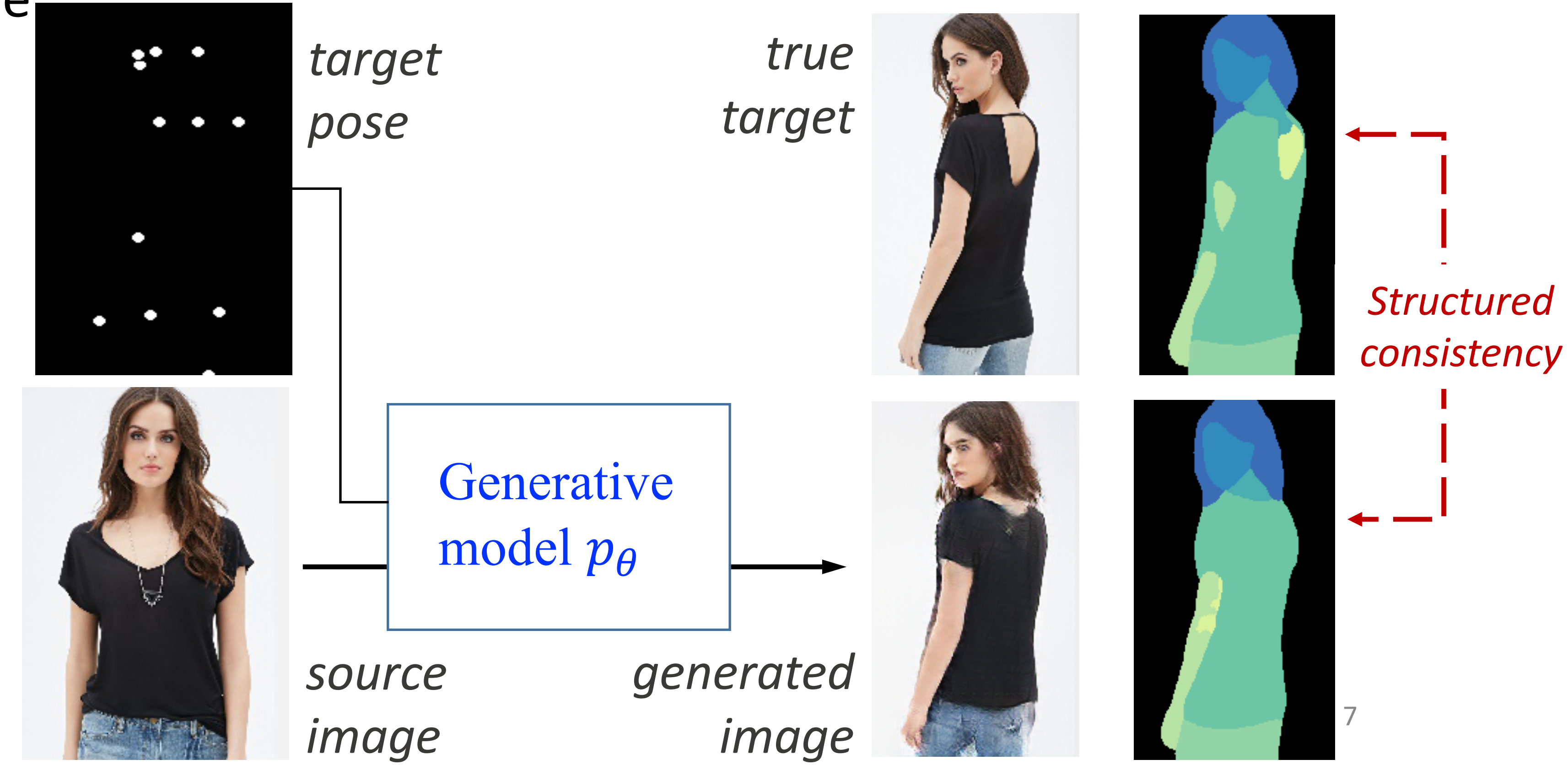
Generative model p_θ

source image

generated image

Difficulty in exploit problem structures and domain knowledge

- Pose Conditional Person Image Generation
  - Given a person image and a target pose, generate an image of the person under the new pose



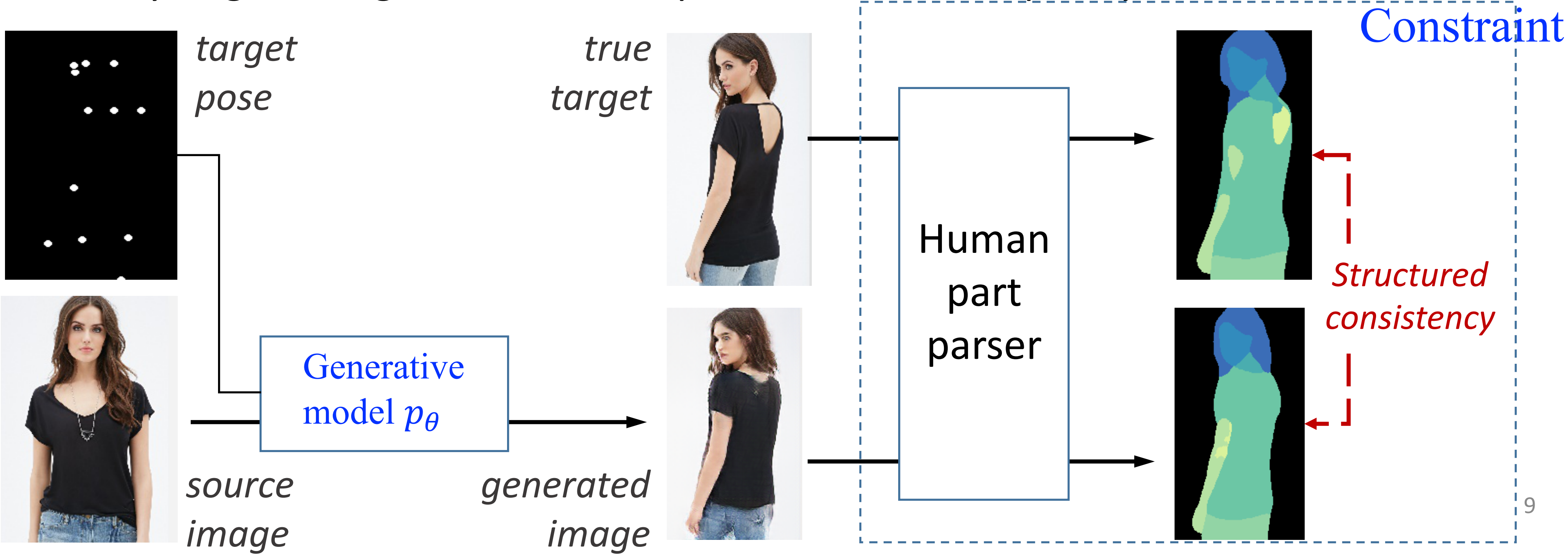
- Generative models can be trained on **supervised data**
- But fail to use **structured knowledge**, i.e., human body structure
  - head, main body, arms, …

Existing approaches to adding structured knowledge

- Designing specialized neural architectures to hard-code knowledge
  - E.g., ConvNets conv-pooling architecture: translation-invariance of image classification
  - Usually only applicable to specific knowledge, models, or tasks

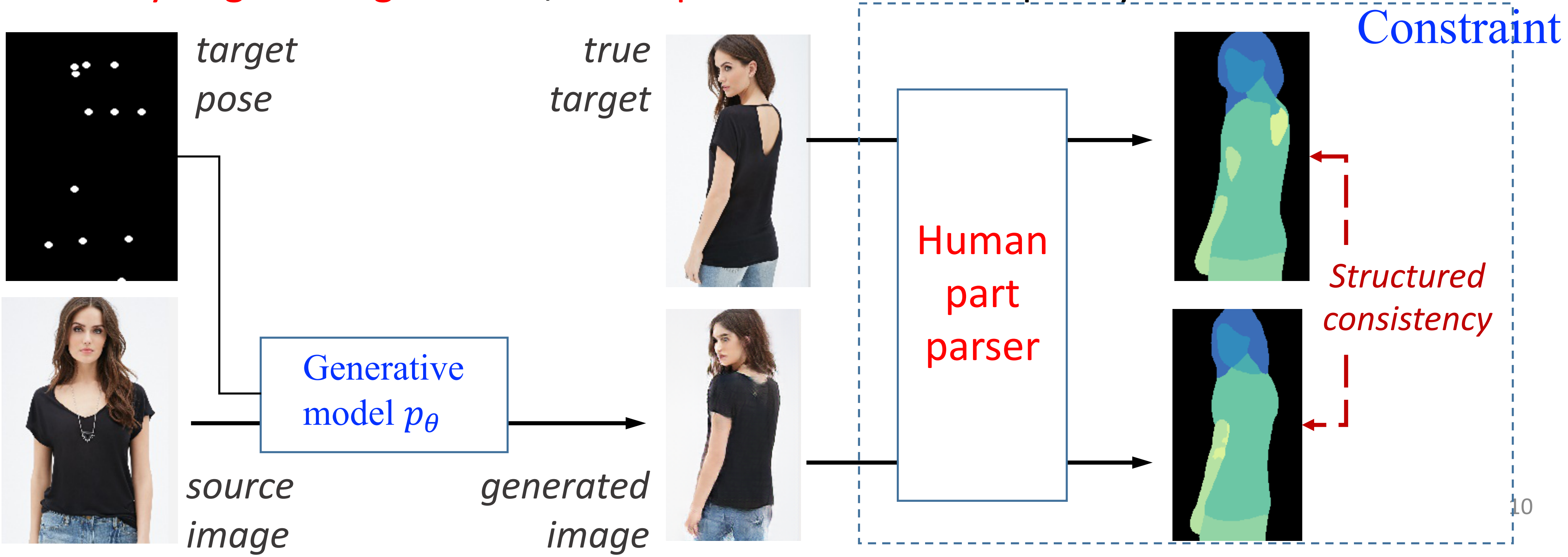
Existing approaches to adding structured knowledge

- Posterior regularization (PR) [Ganchev et al., 10; Hu et al., 16]
  - Imposes knowledge constraints on posterior distributions of probabilistic models
  - Many DGMs lack probabilistic Bayesian formulation / meaningful latent variables
  - Require *a priori* fixed constraints
    - Heavy engineering burden / sub-optimal without adaptivity to the data and models



Existing approaches to adding structured knowledge

- Posterior regularization (PR) [Ganchev et al., 10; Hu et al., 16]
    - Imposes knowledge constraints on posterior distributions of probabilistic models
    - Many DGMs lack probabilistic Bayesian formulation / meaningful latent variables
    - Require *a priori* fixed constraints
        - Heavy engineering burden / sub-optimal without adaptivity to the data and models

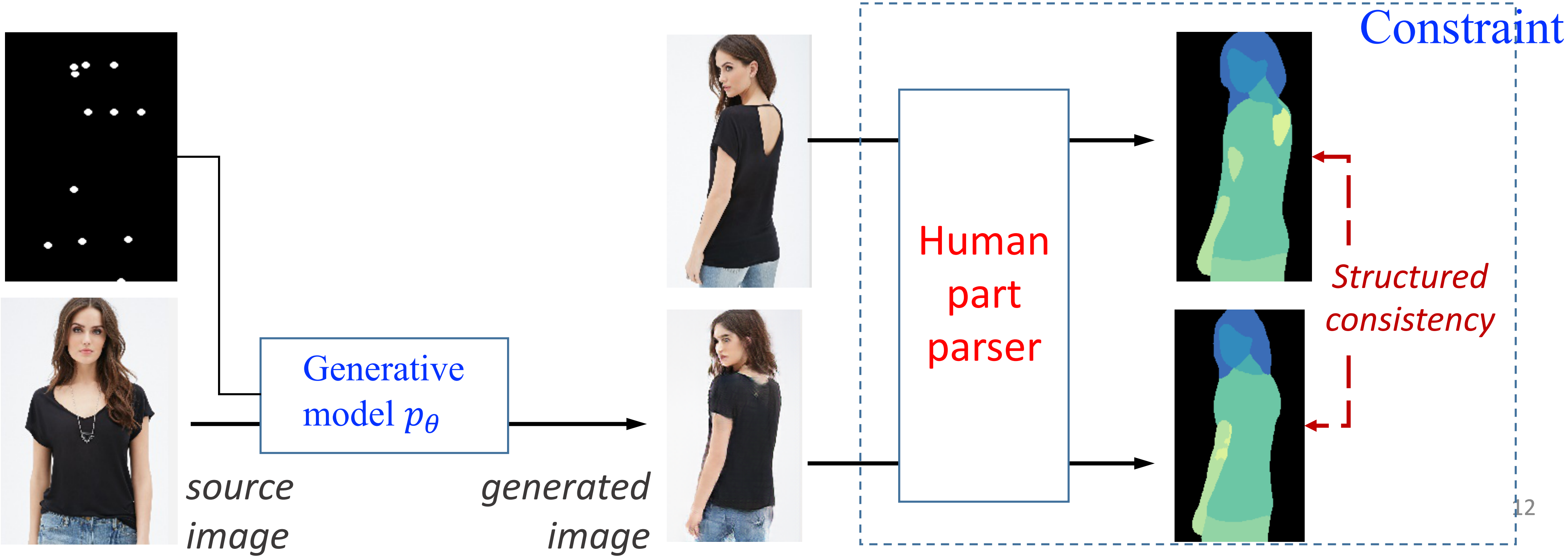


This work: DGMs with learnable knowledge

- 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

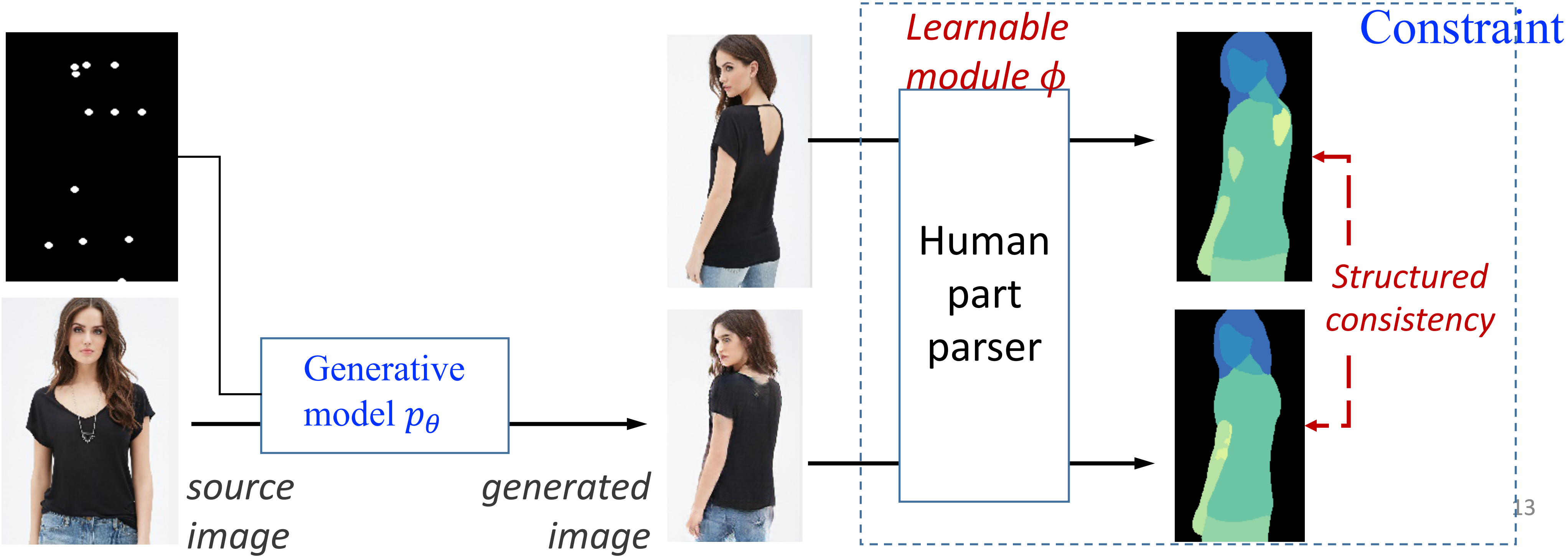
This work: DGMs with learnable knowledge

- 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



This work: DGMs with learnable knowledge

- 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



Posterior Regularization for DGMs

- Consider a generative model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathrm{R}$
  - higher f value, better $\boldsymbol{x}$ w.r.t. the knowledge
- PR assumes a variational distribution q, and the objective

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

  - Solve with an EM-style procedure

E-step: $q^*(\boldsymbol{y}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x})\exp\{ \alpha f(\boldsymbol{x}) \}$

M-step: $\min_\theta \mathrm{KL}\big(q^*(\boldsymbol{x}) || p_\theta(\boldsymbol{x})\big) = \min_\theta -\mathbb{E}_{q^*} [\log p_\theta(\boldsymbol{x})] + const.$

Posterior Regularization for DGMs

vanilla PR: constraint f is fixed *a priori*

- Consider a generative model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathrm{R}$
  - higher f value, better $\boldsymbol{x}$ w.r.t. the knowledge
- PR assumes a variational distribution q, and the objective

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

  - Solve with an EM-style procedure

E-step: $q^*(\boldsymbol{y}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x}) \exp\{ \alpha f(\boldsymbol{x}) \}$

M-step: $\min_\theta \mathrm{KL}\big(q^*(\boldsymbol{x}) || p_\theta(\boldsymbol{x})\big) = \min_\theta -\mathbb{E}_{q^*} [\log p_\theta(\boldsymbol{x})] + const.$

Posterior Regularization for DGMs

vanilla PR: constraint f is fixed *a priori*

- Consider a generative model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f_\phi(\boldsymbol{x}) \in \mathrm{R}$
  - higher f value, better $\boldsymbol{x}$ w.r.t. the knowledge

We want to allow learnable components, i.e. f_ϕ

- PR assumes a variational distribution q, and the objective

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

  - Solve with an EM-style procedure

E-step: $q_\phi(\boldsymbol{y}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x})\exp\{\, \alpha f_\phi(\boldsymbol{x})\,\}$

M-step: $\min_\theta \mathrm{KL}\left(q_\phi(\boldsymbol{x})||p_\theta(\boldsymbol{x})\right) = \min_\theta -\mathbb{E}_{q_\phi}\,[\log p_\theta(\boldsymbol{x})] + const.$

Posterior Regularization for DGMs

vanilla PR: constraint f is fixed *a priori*

- Consider a generative model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f_\phi(\boldsymbol{x}) \in \mathrm{R}$
  - higher f value, better $\boldsymbol{x}$ w.r.t. the knowledge

We want to allow learnable components, i.e. f_ϕ

- PR assumes a variational distribution q, and the objective

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

  - Solve with an EM-style procedure

E-step: $q_\phi(\boldsymbol{y}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x})\exp\{\, \alpha f_\phi(\boldsymbol{x}) \,\}$

M-step: $\min_\theta \mathrm{KL}\left(q_\phi(\boldsymbol{x}) || p_\theta(\boldsymbol{x})\right) = \min_\theta -\mathbb{E}_{q_\phi} \left[\log p_\theta(\boldsymbol{x})\right] + const.$

Optimize $\boldsymbol{\phi}$ *in M-step:* $\max_\phi \mathbb{E}_{\boldsymbol{x}\sim q}[f_\phi(\boldsymbol{x})]$

problematic as the generated samples $\boldsymbol{x} \sim q$ can be of low quality

Entropy-Regularized Policy Optimization (ERPO)

- ERPO:
  - Policy gradient with information theoretic regularizers
  - E.g., Relative Entropy Policy Search [Peters et al., 2010]
- **In RL convention:** assume state s, action a, policy $p_\pi(a|s)$, reward $R(s,a) \in \mathrm{R}$; μ^π is the stationary state distribution
- **To map to PR:** Let $\boldsymbol{x} = (s,a)$, $p_\pi(\boldsymbol{x}) = \mu^\pi(s)p_\pi(a|s)$
- Let $q_\pi(\boldsymbol{x})$ be policy at iteration t and $p_\pi(\boldsymbol{x})$ at iteration $t-1$
- Objective

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

Close resemblance b/w PR and ERPO

- PR $\min_{\theta, q} \mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || \, p_\theta(\boldsymbol{x})) - \alpha \, \mathbb{E}_q\left[f_\phi(\boldsymbol{x})\right]$
- ERPO $\min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x}) || \, p_\pi(\boldsymbol{x})) - \alpha \, \mathbb{E}_q[R(\boldsymbol{x})]$

PR		**ERPO**
Generative model $p_\theta(\boldsymbol{x})$	$\longleftrightarrow$	Reference (old) policy $p_\pi(\boldsymbol{x})$
Constraint $f_\phi(\boldsymbol{x})$	$\longleftrightarrow$	Reward $R(\boldsymbol{x})$
$q_\phi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x})\mathrm{exp}\left\{\, \alpha f_\phi(\boldsymbol{x}) \,\right\}$	$\longleftrightarrow$	$q_\pi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\pi(\boldsymbol{x})\mathrm{exp}\{\, \alpha R(\boldsymbol{x}) \,\}$

Close resemblance b/w PR and ERPO

- PR $\min_{\theta, q} \mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || \, p_\theta(\boldsymbol{x})) - \alpha \, \mathbb{E}_q\left[f_\phi(\boldsymbol{x})\right]$
- ERPO $\min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x}) || \, p_\pi(\boldsymbol{x})) - \alpha \, \mathbb{E}_q[R(\boldsymbol{x})]$

Inverse reinforcement learning (IRL) can learn a reward function from data

PR		**ERPO**
Generative model $p_\theta(\boldsymbol{x})$	⟷	Reference (old) policy $p_\pi(\boldsymbol{x})$
Constraint $f_\phi(\boldsymbol{x})$	⟷	Reward $R(\boldsymbol{x})$
$q_\phi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x})\exp\left\{ \alpha f_\phi(\boldsymbol{x}) \right\}$	⟷	$q_\pi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\pi(\boldsymbol{x})\exp\{ \alpha R(\boldsymbol{x}) \}$

Close resemblance b/w PR and ERPO

- PR $\quad \min_{\theta, q} \mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || p_\theta(\boldsymbol{x})) - \alpha \, \mathbb{E}_q\left[f_\phi(\boldsymbol{x})\right]$
- ERPO $\quad \min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x}) || p_\pi(\boldsymbol{x})) - \alpha \, \mathbb{E}_q[R(\boldsymbol{x})]$

Inverse reinforcement learning (IRL) can learn a reward function from data

PR		**ERPO**
Generative model $p_\theta(\boldsymbol{x})$	$\longleftrightarrow$	Reference (old) policy $p_\pi(\boldsymbol{x})$
Constraint $f_\phi(\boldsymbol{x})$	$\longleftrightarrow$	Reward $R(\boldsymbol{x})$
$q_\phi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x}) \exp\{\alpha f_\phi(\boldsymbol{x})\}$	$\longleftrightarrow$	$q_\pi^*(\boldsymbol{y}\|\boldsymbol{x}) \propto p_\pi(\boldsymbol{x}) \exp\{\alpha R(\boldsymbol{x})\}$

We can transfer IRL technique to learn $f_\phi(\boldsymbol{x})$!

Maximum-Entropy Inverse Reinforcement Learning (MaxEnt IRL)

- ERPO $\min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x}) || \, p_\pi(\boldsymbol{x})) - \alpha \, \mathbb{E}_q[R(\boldsymbol{x})]$

$$q_\pi^*(\boldsymbol{y}|\boldsymbol{x}) \propto p_\pi(\boldsymbol{x}) \exp\{ \, \alpha R(\boldsymbol{x}) \, \}$$

- MaxEnt IRL learns reward function $R_\phi(\boldsymbol{x})$ with unknown parameters $\boldsymbol{\phi}$
- Assumes reference policy $p_\pi(\boldsymbol{x})$ as a uniform, so the above KL regularization becomes an entropy regularization (i.e., MaxEnt)
- The above $q_\pi^*(\boldsymbol{y}|\boldsymbol{x})$ now additionally depends on $\boldsymbol{\phi}$

$$q_{\pi,\phi}(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\left\{ \, \alpha R_\phi(\boldsymbol{x}) \, \right\}$$

- Learns $\boldsymbol{\phi}$ by maximizing data log-likelihood

$$\boldsymbol{\phi}^* = \mathrm{argmax}_\phi \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log q_{\pi,\phi}(\boldsymbol{x})]$$

Maximum-Entropy Inverse Reinforcement Learning (MaxEnt IRL)

- ERPO $\min_{q_\pi} \mathcal{L}(q_\pi) = \mathrm{KL}(q_\pi(\boldsymbol{x}) || \, p_\pi(\boldsymbol{x})) - \alpha \, \mathbb{E}_q[R(\boldsymbol{x})]$

  $$q_\pi^*(\boldsymbol{y}|\boldsymbol{x}) \propto p_\pi(\boldsymbol{x}) \mathrm{exp}\{ \, \alpha R(\boldsymbol{x}) \, \}$$

- MaxEnt IRL learns reward function $R_\phi(\boldsymbol{x})$ with unknown parameters $\boldsymbol{\phi}$
- Assumes reference policy $p_\pi(\boldsymbol{x})$ as a uniform, so the above KL regularization becomes an entropy regularization (i.e., MaxEnt)
- The above $q_\pi^*(\boldsymbol{y}|\boldsymbol{x})$ now additionally depends on $\boldsymbol{\phi}$

  $$q_{\pi,\phi}(\boldsymbol{y}|\boldsymbol{x}) \propto \mathrm{exp}\big\{ \, \alpha R_\phi(\boldsymbol{x}) \, \big\}$$

- Learns $\boldsymbol{\phi}$ by maximizing data log-likelihood

  $$\boldsymbol{\phi}^* = \mathrm{argmax}_\phi \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log q_{\pi,\phi}(\boldsymbol{x})]$$

Apply this objective to learn f_ϕ in PR

Algorithm: PR with learnable constraint

- PR: $\min_{\theta,q} \mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || \, p_\theta(\boldsymbol{x})) - \alpha \, \mathbb{E}_q\left[f_\phi(\boldsymbol{x})\right]$
- **(1) Learning the constraint f_ϕ**

  E-step: $q_\phi(\boldsymbol{y}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x}) \exp\{\, \alpha f_\phi(\boldsymbol{x}) \,\}$

  - Use the same objective as in MaxEnt IRL

  $$\boldsymbol{\phi}^* = \operatorname{argmax}_\phi \mathbb{E}_{\boldsymbol{x} \sim p_{data}}\left[\log q_\phi(\boldsymbol{x})\right]$$

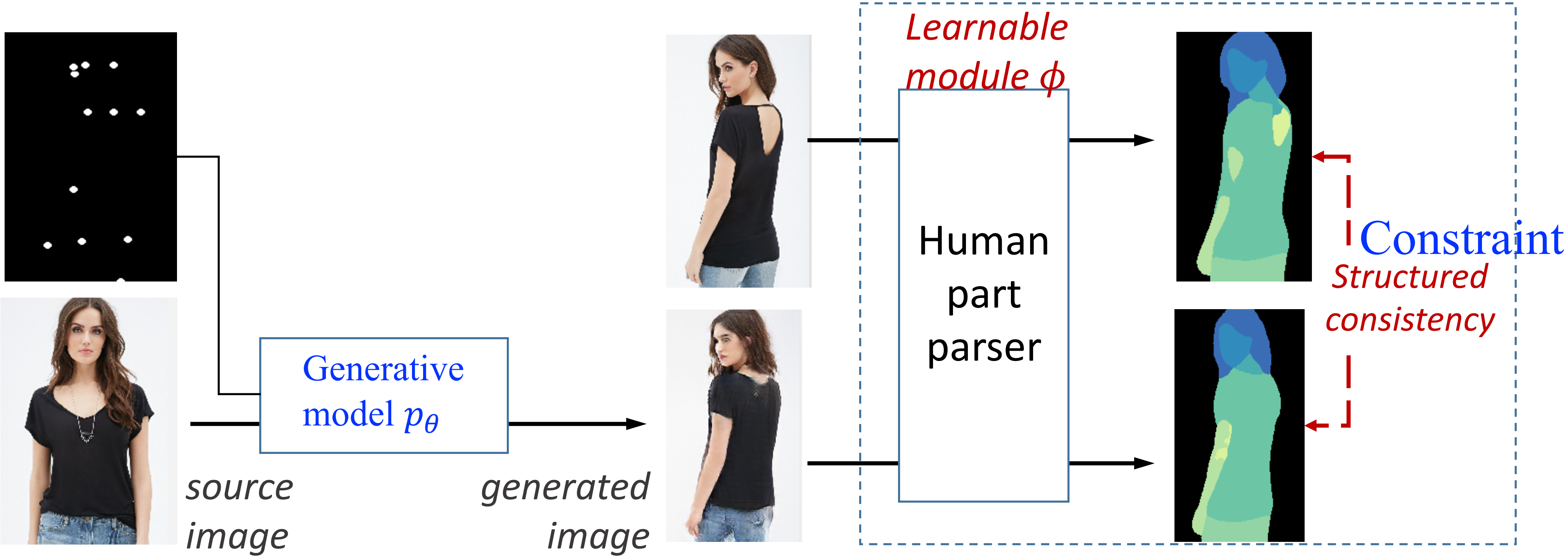
- **(2) Learning the generative model p_θ**

  M-step: $\min_\theta \mathrm{KL}\left(q_\phi(\boldsymbol{x}) || p_\theta(\boldsymbol{x})\right) = \ \min_\theta -\mathbb{E}_{q_\phi} \left[\log p_\theta(\boldsymbol{x})\right] + const.$

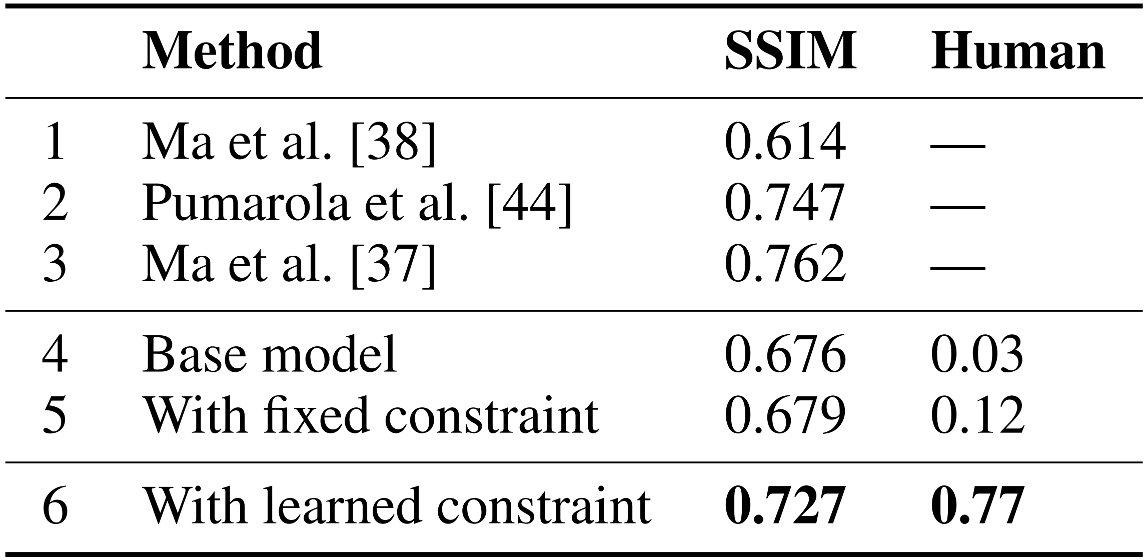
  - If p_θ is an *implicit* model (e.g., GANs), we can approximate by minimizing reverse KL

  $$\min_\theta \mathrm{KL}\left(p_\theta(\boldsymbol{x}) || q_\phi(\boldsymbol{x})\right)$$

Experiments – Pose-conditional Human Image Generation



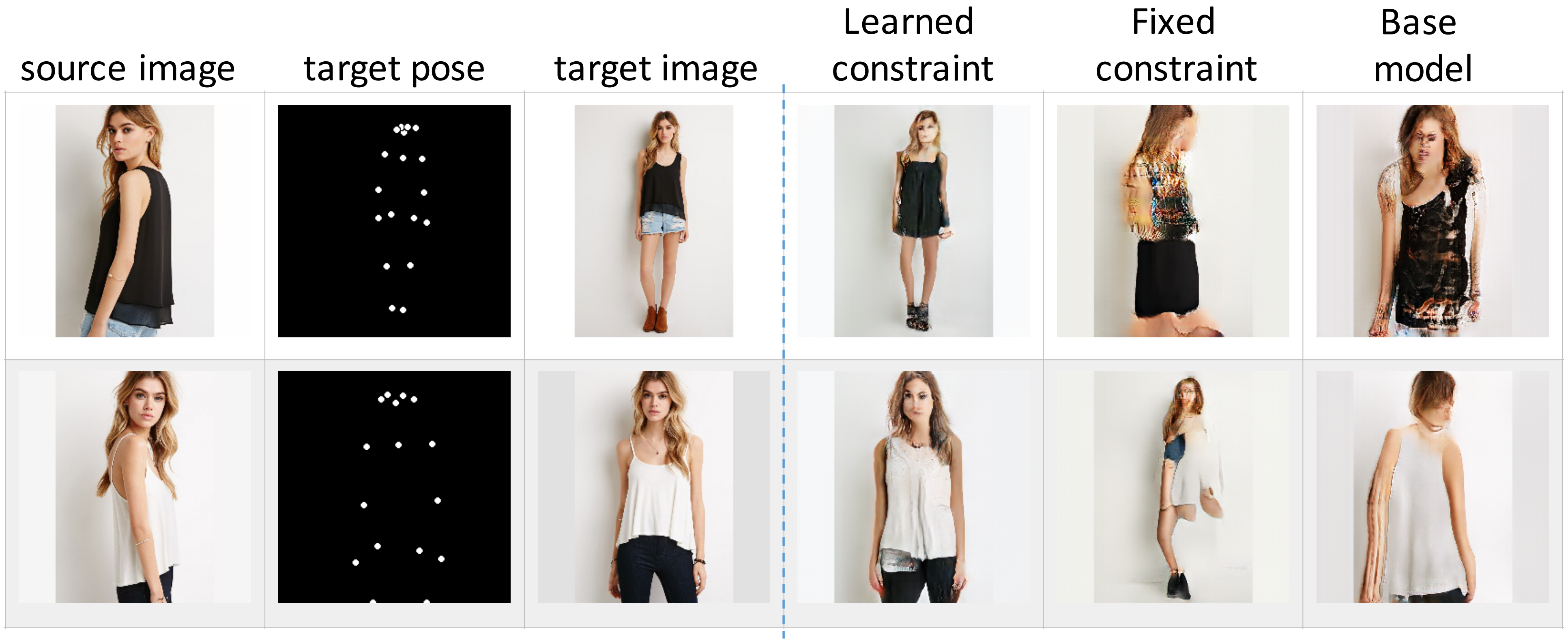
Experiments – Pose-conditional Human Image Generation



	Method	SSIM	Human
1	Ma et al. [38]	0.614	—
2	Pumarola et al. [44]	0.747	—
3	Ma et al. [37]	0.762	—
4	Base model	0.676	0.03
5	With fixed constraint	0.679	0.12
6	With learned constraint	**0.727**	**0.77**

Results of image generation on Structural Similarity (SSIM) between generated and true images, and human survey where the full model yields better generations than the base models (Rows 5-6) on 77% test cases.

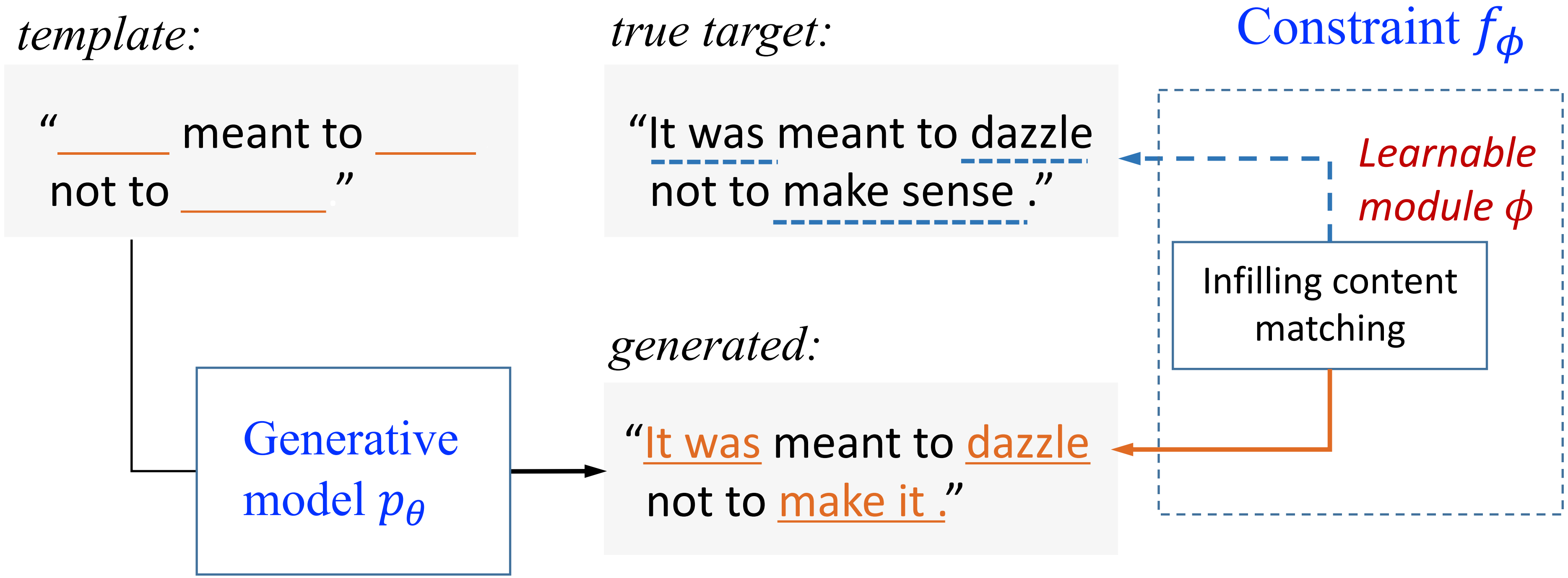
Experiments – Pose-conditional Human Image Generation



Samples generated by the models. The model with learned human part constraint generates correct poses and preserves human body structure much better.

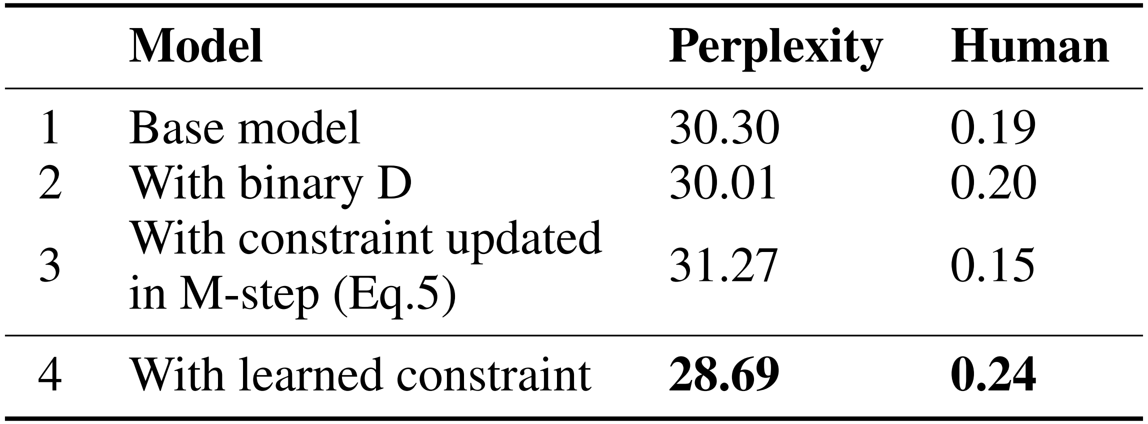
Experiments – Template-guided Sentence Generation

- **Task:** Given a template, generate a complete sentence following the template
- **Constraint:** force the match between the infilling content of the generated sentence with the true content



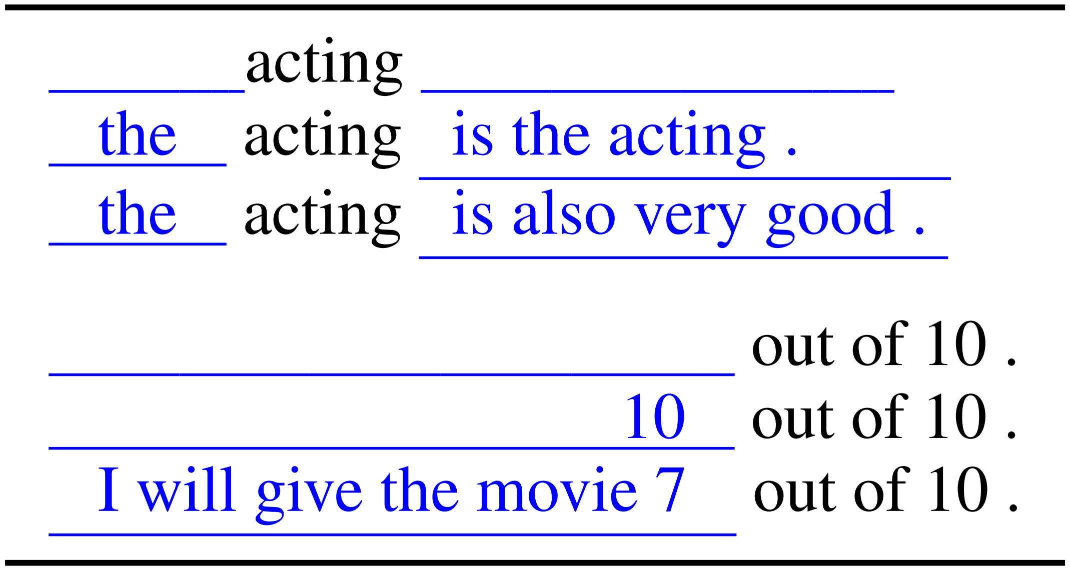
template:
“_____ meant to _____ not to ______”
true target:
“It was meant to dazzle not to make sense .”
Constraint f_ϕ
Learnable module ϕ
Infilling content matching
generated:
“It was meant to dazzle not to make it .”
Generative model p_θ

Experiments – Template-guided Sentence Generation



	Model	Perplexity	Human
1	Base model	30.30	0.19
2	With binary D	30.01	0.20
3	With constraint updated in M-step (Eq.5)	31.27	0.15
4	With learned constraint	**28.69**	**0.24**

Samples by the full model are considered as of higher quality in 24% cases.



______ acting ____________
the acting is the acting .
the acting is also very good .

________________ out of 10 .
10 out of 10 .
I will give the movie 7 out of 10 .

Two test examples, including the template, the sample by the base model, and the sample by the constrained model.

Conclusions

- Formal connections between posterior regularization (PR) and reinforcement learning (RL)
- Learn the knowledge constraints in PR as reward learning in (inverse) RL
- The resulting algorithm is:
  - generally applicable to any deep generative models
  - flexible to learn the constraints and model jointly
- Experiments on image and text generation showed the effectiveness of the algorithm