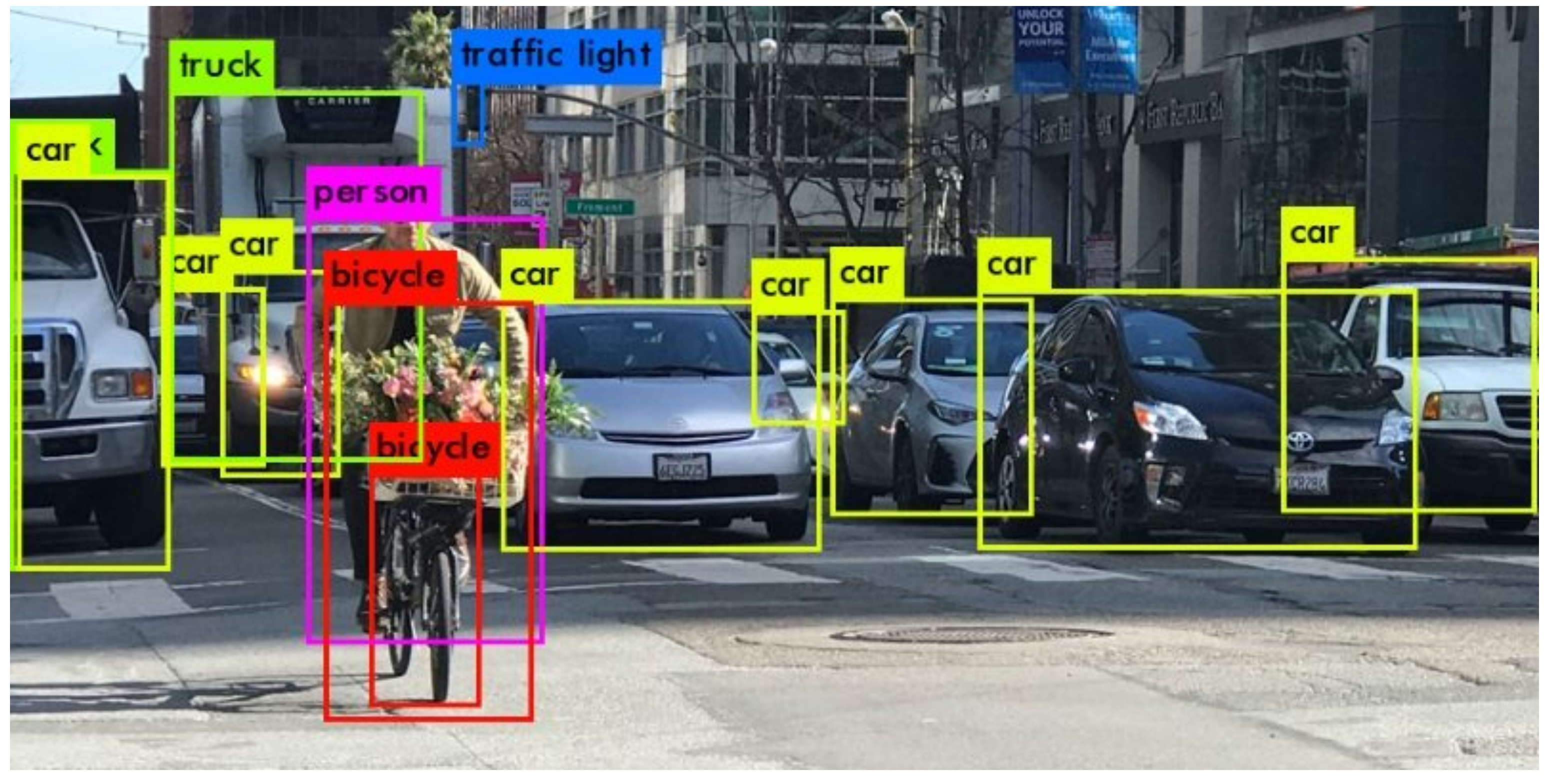
Learning with ALL Experiences

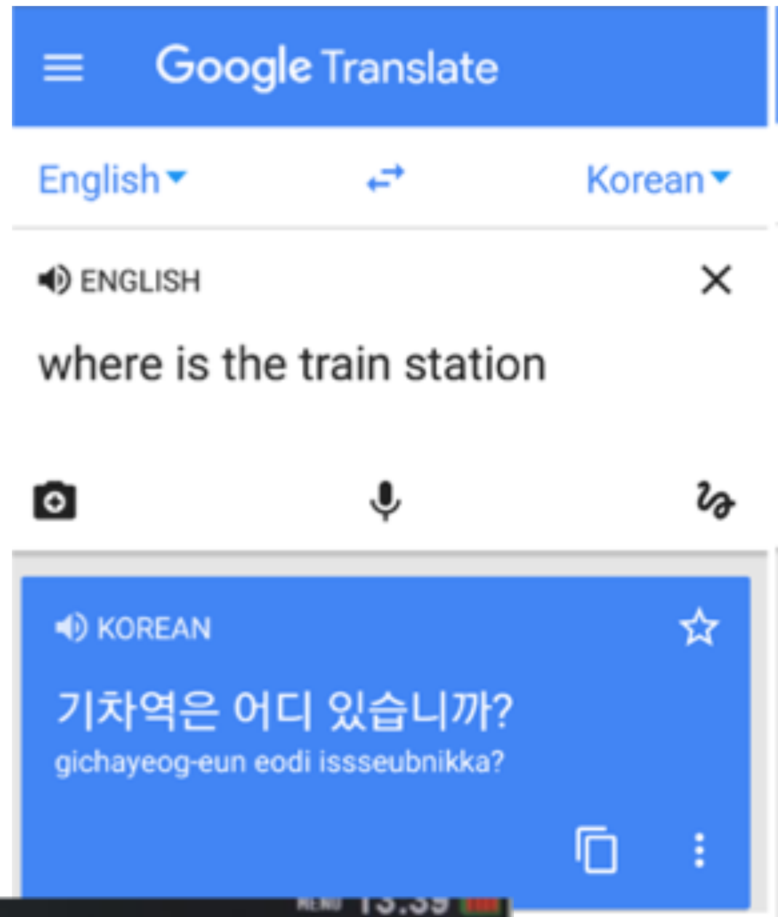
—— A Standardized ML Formalism

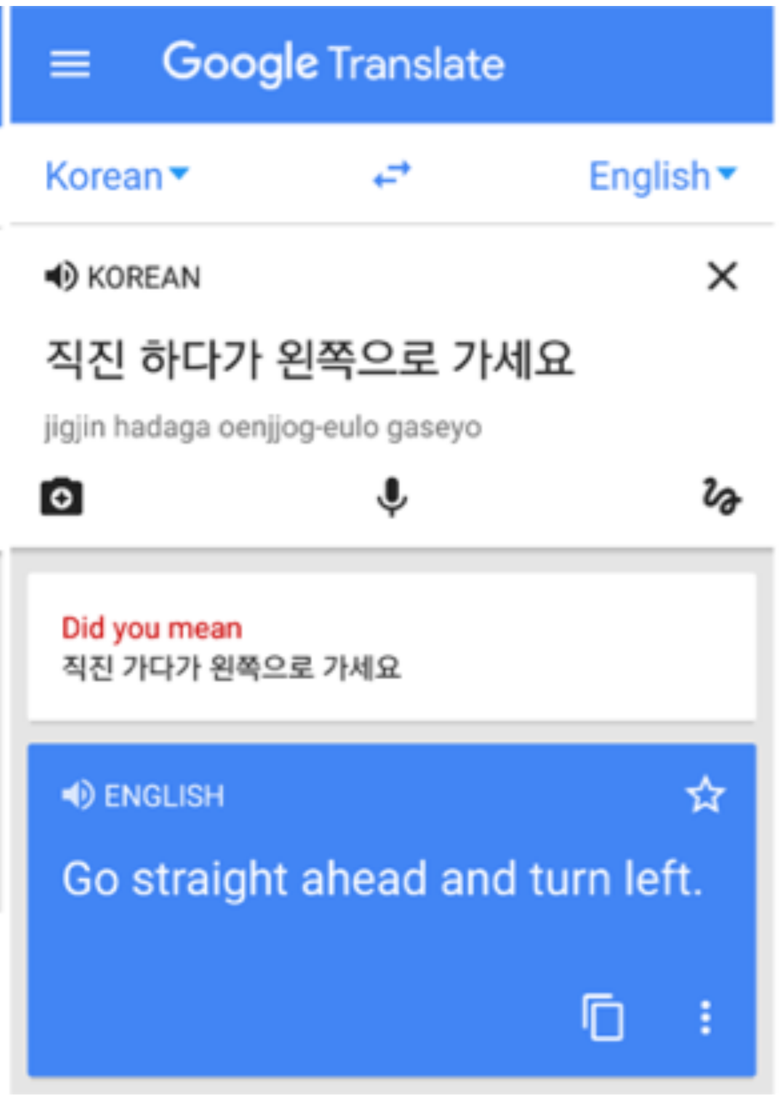
Zhiting Hu
Assistant Professor
HDSI, UCSD

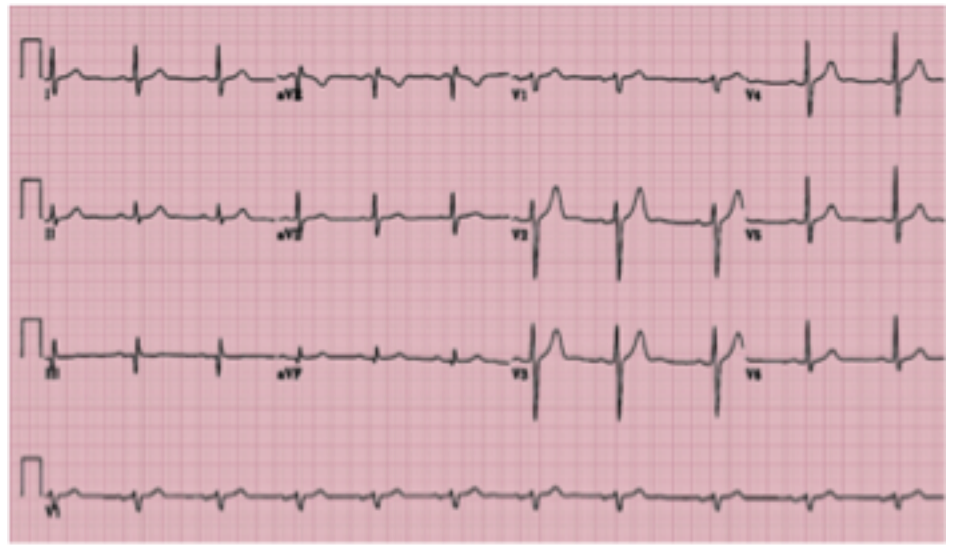
UC San Diego
Carnegie Mellon University

Real-world Machine Learning Problems





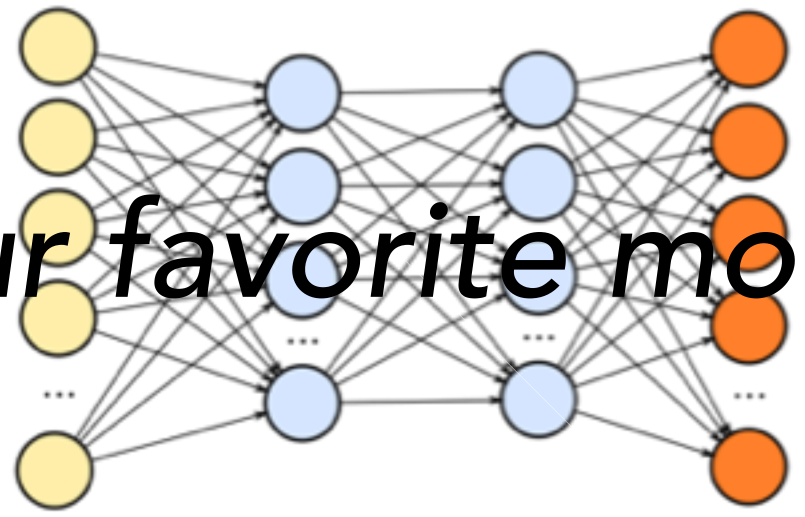




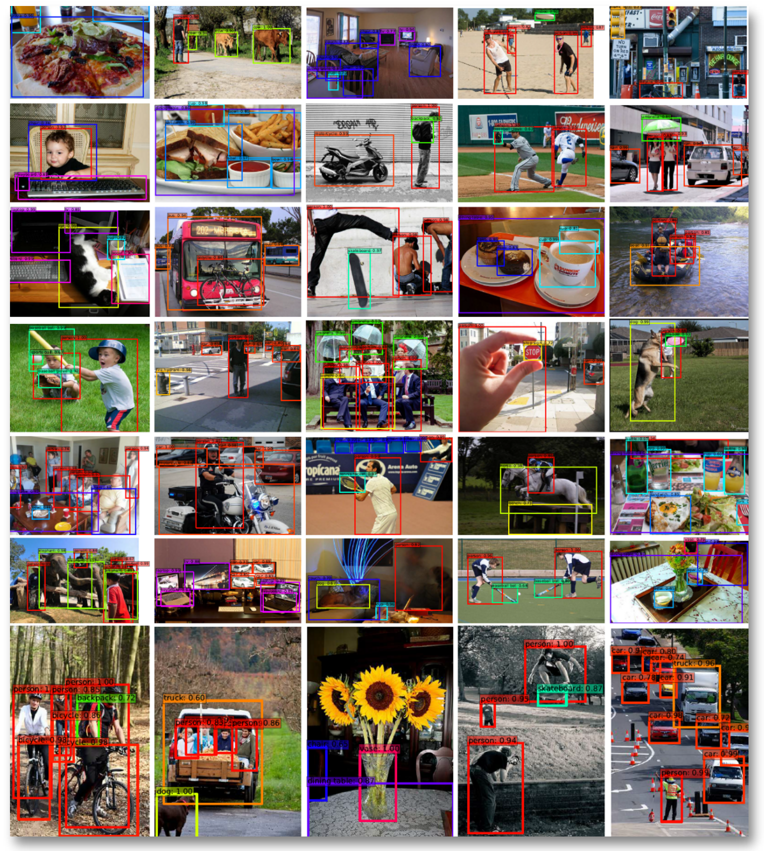
Machine learning: computational methods that enable machines to ***learn from experiences****.*

supervision signals that drive the learning

Experiences → *Your favorite model*



Experiences of all kinds



Data examples

Type-2 diabetes is 90% more common than type-1

Constraints

Rewards

Auxiliary agents

Adversaries

…

And all combinations of of that …

How human beings solve them ALL?

The zoo of algorithms and heuristics

maximum likelihood estimation

reinforcement learning as inference

data re-weighting

inverse RL

policy optimization

active learning

data augmentation

actor-critic

reward-augmented maximum likelihood

label smoothing

imitation learning

softmax policy gradient

adversarial domain adaptation

posterior regularization

GANs

constraint-driven learning

knowledge distillation

intrinsic reward

generalized expectation

prediction minimization

regularized Bayes

learning from measurements

energy-based GANs

weak/distant supervision

To solve problems by integrating all possible sources of information

How can we design learning systems that can learn from all types of experiences?

Example Problems: Controllable Content Generation

Example Problems: Controllable Content Generation

- Ex1: Text

Applications: personalized chatbot, live sports commentary production

Controlling ser

Pos *The film is full o*

Neg *The film is st*

- Sentiment classifier
- Linguistic rules

…

ng writing style

nes contributed 26 points, 8
7 assists.

Elaborate

nes rounded out the box score
around impressive performance,
scoring 26 points, grabbing 8 rebounds
and dishing out 7 assists.

[Hu et al., ICML'17]

[Lin et al., EMNLP'20Findings]

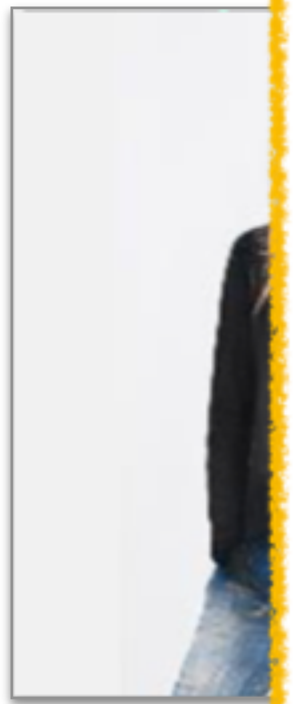
Example Problems: Controllable Content Generation

Ex1: Text

Ex2: Fashion images

Applications: virtual clothing try-on system

Source image



Constraints of human gesture

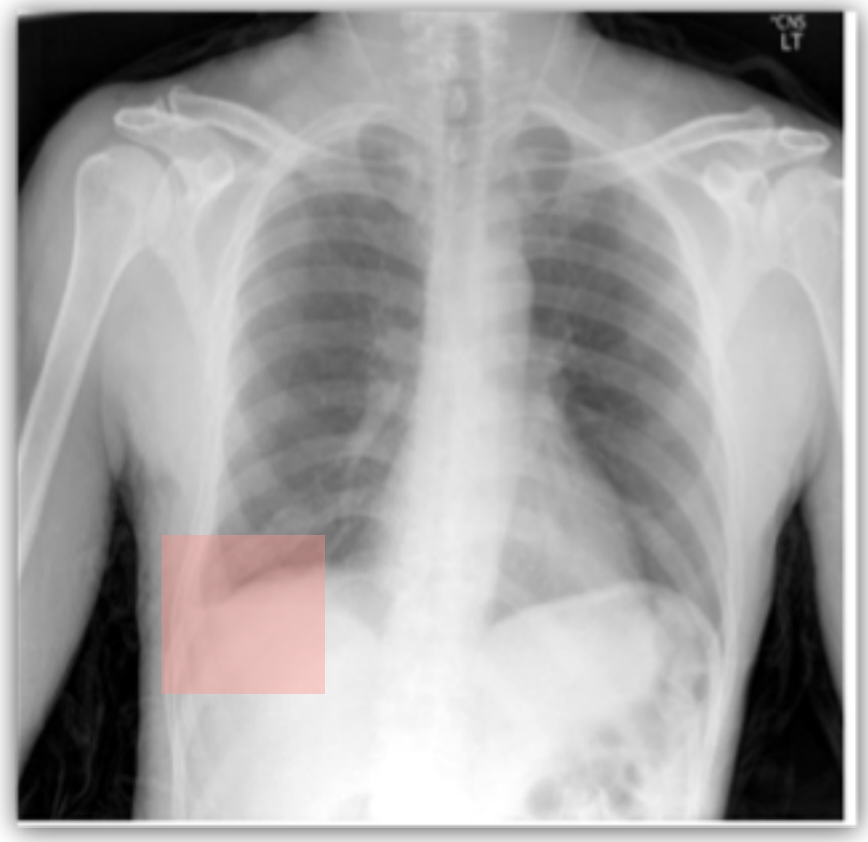
nt poses

[Hu et al., NeurIPS’18]

Example Problems: Controllable Content Generation

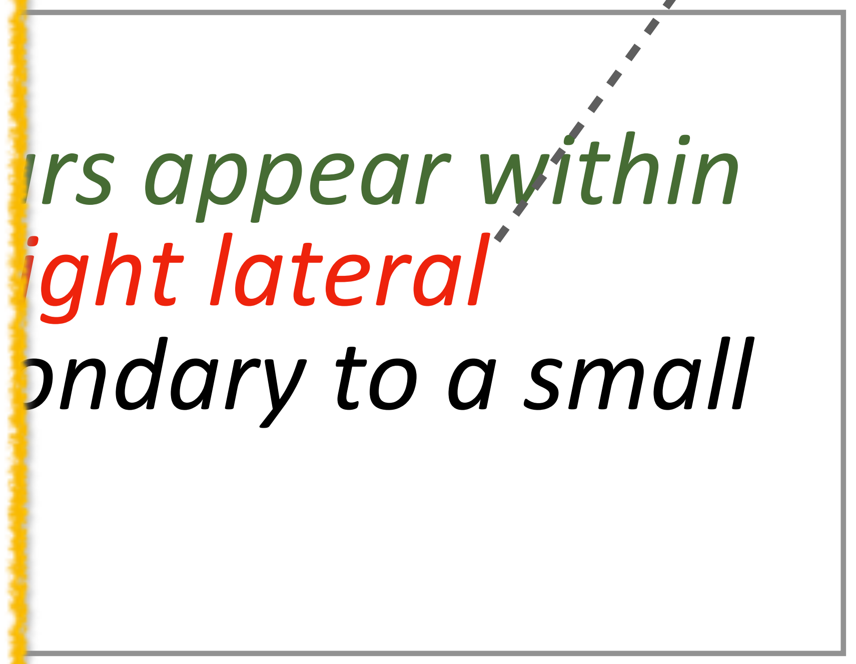
Ex1: Text Ex2: Fashion images Ex3: Medical reports

Applications: assistive diagnosis



- Medical domain knowledge
- Expert feedbacks

Abnormal findings



[Li et al., NeurIPS'18, AAAI'19]

Example Problems: Controllable Content Generation

Ex1: Text Ex2: Fashion images Ex3: Medical reports

How can we design learning systems that can learn from all types of experiences?

Limitations of the dominant way for designing learning systems

- Isolated learning frameworks

  A bewildering marketplace hard to navigate through

maximum likelihood estimation
reinforcement learning as inference
data re-weighting
inverse RL
policy optimization
active learning
data augmentation
actor-critic
reward-augmented maximum likelihood
label smoothing
imitation learning
softmax policy gradient
adversarial domain adaptation
posterior regularization
GANs
constraint-driven learning
knowledge distillation
intrinsic reward
generalized expectation
prediction minimization
regularized Bayes
learning from measurements
energy-based GANs
weak/distant supervision

Limitations of the dominant way for designing learning systems

- Isolated learning frameworks
  A bewildering marketplace hard to navigate through

- Isolated problem solving
  Choose/adapt algorithms for particular set of experiences

Limitations of the dominant way for designing learning systems

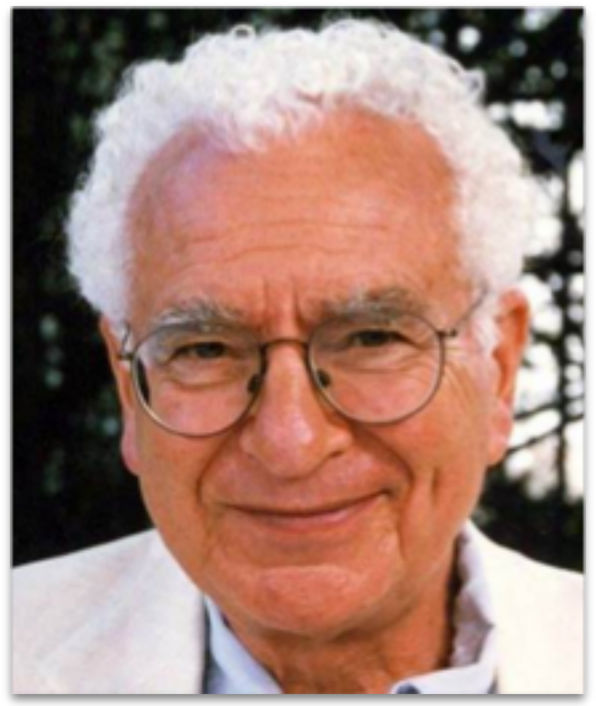
- Isolated learning frameworks
  A bewildering marketplace hard to navigate through

- Isolated problem solving
  Choose/adapt algorithms for particular set of experiences

- Isolated algorithm innovations
  E.g. better learning-from-reward rarely informs better learning-from-constraints

An alternative way …

Need a unifying perspective



You don't need something more in order to get something more.

-- Murray Gell-Mann (1929–2019), Physicist, Nobel laureate

An alternative way under a unifying perspective

- Isolated learning frameworks
- A standardized ML formalism

  A holistic understanding of the diverse ML techniques

- Isolated problem solving
- Integrated problem solving

  Plugging in arbitrary available experiences to drive learning

- Isolated algorithm innovations
- Integrated algorithm innovation

  E.g. automated **reward** acquisition —> automated **constraint** inducing

  —> automated **data** augmentation

  stabilized **policy** training —> stabilized **GAN** training

An alternative way under a unifying perspective

- A standardized ML formalism
  A holistic understanding of the diverse ML techniques

- Integrated problem solving
  Plugging in arbitrary available experiences to drive learning

- Integrated algorithm innovation
  E.g. automated **reward** acquisition —> automated **constraint** inducing
  —> automated **data** augmentation
  stabilized **policy** training —> stabilized **GAN** training

Outline

1. A standardized ML formalism
2. Integrated problem solving
3. Integrated algorithm innovation

Outline

1. A standardized ML formalism
2. Integrated problem solving
3. Integrated algorithm innovation

A standardized ML formalism

$$\min_{\color{green}{q}, \theta} \ \mathbb{E} - \mathbb{D} - \mathbb{H}$$

Experience Divergence Uncertainty

MLE at a close look:

- The most classical learning algorithm

- Supervised:
  - Observe data $\mathcal{D} = \{(\boldsymbol{x}^*, \boldsymbol{y}^*)\}$
  - Solve with SGD

$$\min_{\theta} - \mathbb{E}_{(\boldsymbol{x}^*, \boldsymbol{y}^*) \sim \mathcal{D}} \left[\log p_\theta(\boldsymbol{y}^* | \boldsymbol{x}^*) \right]$$

- Unsupervised:
  - Observe $\mathcal{D} = \{(\boldsymbol{x}^*)\}$, $\boldsymbol{y}$ is latent variable
  - Posterior $p_\theta(\boldsymbol{y}|\boldsymbol{x})$
  - Solve with EM, etc

$$\min_{\theta} - \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}} \left[\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) \right]$$

MLE as entropy maximization

- Duality between Supervised MLE and maximum entropy, when p is exponential family

Shannon entropy H

$$\min_{p(\boldsymbol{x},\boldsymbol{y})} \; H(p)$$

features $T(\boldsymbol{x}, \boldsymbol{y})$

$$s.t.\; \mathbb{E}_p[T(\boldsymbol{x}, \boldsymbol{y})] = \mathbb{E}_{(x^*,y^*)\sim\mathcal{D}}[T(\boldsymbol{x}, \boldsymbol{y})]$$

data as constraints

** Proof with Lagrangian method*

MLE as entropy maximization

- Unsupervised MLE can be achieved by maximizing the negative free energy:
  - Introduce auxiliary distribution $q(\boldsymbol{y}|\boldsymbol{x})$ (and then play with its entropy and cross entropy, etc.)

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}\left[\log \frac{p_\theta(\boldsymbol{x}^*, \boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^*)}\right] + \mathrm{KL}\big(q(\boldsymbol{y}|\boldsymbol{x}^*) \,||\, p_\theta(\boldsymbol{y}|\boldsymbol{x}^*)\big)$$

$$\geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})] \coloneqq \mathcal{L}(q, \boldsymbol{\theta})$$

Solvers for unsupervised MLE

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) \geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})] \coloneqq \mathcal{L}(q, \boldsymbol{\theta})$$

Solve with **EM**

- E-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t q, equivalent to minimizing KL by setting
  $$q(\boldsymbol{y}|\boldsymbol{x}^*) = p_{\boldsymbol{\theta}^{old}}(\boldsymbol{y}|\boldsymbol{x}^*)$$
- M-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$: $\max_\theta \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$

When $p_{\boldsymbol{\theta}}$ is complex, solve with **variational EM**

When q is also complex, solve with **wake-sleep** [Hinton et al., 1995], **VAE**, etc

Posterior Regularization (PR)

- Make use of constraints in Bayesian learning [Ganchev et al., 2010; Zhu et al., 2014]
- Generalized to more general learning settings [Hu et al., 2016]
  - E.g., Complex rule constraints on general NNs
  - Auxiliary distribution $q(\boldsymbol{x}, \boldsymbol{y})$
  - Constant weight $\alpha = \beta > 0$, Slack variable ξ

$$\min_{q,\theta,\xi} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] + \xi$$

$$s.t.\ \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[1 - r(\boldsymbol{x}, \boldsymbol{y}) \right] \leq \xi$$

E.g., $r(\boldsymbol{x}, \boldsymbol{y})$ is a 1st-order logic rule:

Sentence $\boldsymbol{x}$: *A* but *B*
⇒ its sentiment $\boldsymbol{y}$ is the sentiment of *B*

"This was a terrific movie, but the director could have done better."

EM for the general PR

- Rewrite without slack variable:

$$\min_{q,\theta} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

- Solve with EM
  - E-step: $q(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{ \dfrac{\beta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{x}, \boldsymbol{y})}{\alpha} \right\} / Z$
  - M-step: $\min_{\theta} \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$

Reformulating unsupervised MLE with PR

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) \geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

- Introduce arbitrary $q(\boldsymbol{y}|\boldsymbol{x})$

$$\min_{q,\,\theta,\,\xi} \; -\alpha H(q) - \beta \mathbb{E}_q\Big[\log p_\theta(\boldsymbol{x}, \boldsymbol{y})\Big] + \xi$$

$$s.t.\; -\mathbb{E}_q\Big[f(\boldsymbol{x}\,;\,\mathcal{D})\Big] < \xi$$

Data as constraint.
Given $\boldsymbol{x} \sim \mathcal{D}$, this constraint doesn't influence the solution of q and $\boldsymbol{\theta}$

- $f(\boldsymbol{x}\,;\mathcal{D}) := \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}}[\, \mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x})\,]$
  - A constraint saying $\boldsymbol{x}$ must equal to one of the true data points
  - Or alternatively, the (log) expected similarity of $\boldsymbol{x}$ to dataset $\mathcal{D}$, with $\mathbb{1}(\cdot)$ as the similarity measure (we'll come back to this later)
- $\alpha = \beta = 1$

The Standard Equation (SE)

$$\min_{q,\theta,\xi\geq 0} \alpha\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \beta\mathbb{H}(q) + \xi$$

$$s.t.\ -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] < \xi$$

Equivalently:

$$\min_{q,\theta} - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \alpha\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \beta\mathbb{H}(q)$$

3 terms:

Experiences
(exogenous regularizations)
e.g., data examples, rules

Divergence
(fitness)
e.g., Cross Entropy

Uncertainty
(self-regularization)
e.g., Shannon entropy

Textbook
$f(\boldsymbol{x},\boldsymbol{y}|\ .\ )$

Teacher
$q(\boldsymbol{x},\boldsymbol{y})$

Student
$p_\theta(\boldsymbol{x},\boldsymbol{y})$

Uncertainty

SE with data experience -- unsupervised MLE

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q \Big[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \Big] - \mathbb{E}_q \Big[\; f(\boldsymbol{x}, \boldsymbol{y}) \; \Big]$$

$$f := f(\boldsymbol{x}\,;\,\mathcal{D}) = \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}}[\, \mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x}) \,] \qquad \alpha = \beta = 1$$

$$q = q(\boldsymbol{y}|\boldsymbol{x})$$

SE with data experience -- supervised MLE

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[\; f(\boldsymbol{x}, \boldsymbol{y}) \quad \right]$$

$$f := f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) = \log \mathbb{E}_{(\boldsymbol{x}^*, \boldsymbol{y}^*) \sim \mathcal{D}} \left[\mathbb{1}_{(\boldsymbol{x}^*, \boldsymbol{y}^*)}(\boldsymbol{x}, \boldsymbol{y}) \right] \quad \alpha = 1, \beta = \epsilon$$

SE with "oracle data experience" -- active learning

- Have access to a vast pool of unlabeled data instances
- Can select instances (queries) to be labeled by an oracle (e.g., human)

- Experiences:
  - $u(\boldsymbol{x})$ measures *informativeness* of an instance $\boldsymbol{x}$
    - e.g., Uncertainty on $\boldsymbol{x}$, measured by predictive entropy
  - Instances + oracle labels:

$$f(\boldsymbol{x}, \boldsymbol{y} \,;\, Oracle) = \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D},\, \boldsymbol{y}^* \sim \mathrm{Oracle}(\boldsymbol{x}^*)} \left[\mathbb{1}_{(\boldsymbol{x}^*, \boldsymbol{y}^*)}(\boldsymbol{x}, \boldsymbol{y}) \right]$$

SE and Active Learning

$$\min_{q,\theta} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

$f := f(\boldsymbol{x}, \boldsymbol{y}\,;\, Oracle) + u(\boldsymbol{x})$ $\qquad \alpha = 1, \beta = \epsilon$

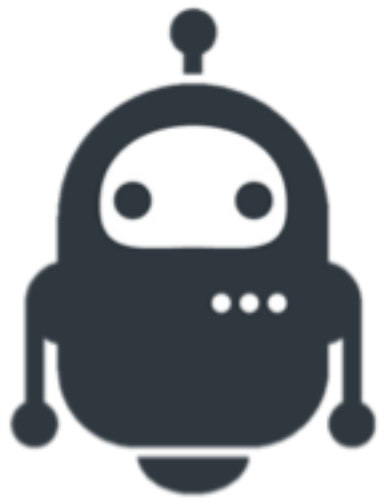
- E-step $\quad q(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{ \dfrac{\beta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{x}, \boldsymbol{y}\,;Oracle) + u(\boldsymbol{x})}{\alpha} \right\} / Z$
- M-step $\quad \min_\theta -\mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$

Equivalent to [e.g., Ertekin et al., 07]:
- Randomly draw a subset $\mathcal{D}_{sub} = \{\boldsymbol{x}^*\}$
- Draw a query $\boldsymbol{x}^*$ from $\mathcal{D}_{sub}$ according to $\exp\{u(\boldsymbol{x})\}$
- Get label $\boldsymbol{y}^*$ for $\boldsymbol{x}^*$ from the oracle
- Maximize log likelihood on $(\boldsymbol{x}^*, \boldsymbol{y}^*)$

SE and Reinforcement learning (RL)

Markov Decision Process (MDP)

AGENT



- State $\boldsymbol{x}_t$
- Take action $\boldsymbol{y}_t \sim p_\theta(\boldsymbol{y}_t|\boldsymbol{x}_t)$

ENVIRONMENT

- Get reward $r_t = r(\boldsymbol{x}_t, \boldsymbol{y}_t)$
- New state $\boldsymbol{x}_{t+1}$

- Policy $p_\theta(\boldsymbol{y}|\boldsymbol{x})$
- $Q^\theta(\boldsymbol{x}, \boldsymbol{y})$ – expected future reward of taking action $\boldsymbol{y}$ in state $\boldsymbol{x}$ and continuing the current policy p_θ

$$Q^\theta(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{p_\theta}\left[\sum_{t=0}^{\infty} r_t \mid \boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{y}_0 = \boldsymbol{y}\right]$$

- $\mu^\theta(\boldsymbol{x})$ – state distribution

$$\mu^\theta(\boldsymbol{x}) = \sum_{t=0}^{\infty} p(\boldsymbol{x}_t = \boldsymbol{x})$$

SE with reward experience I -- RL as inference

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_q \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

- RL-as-inference [Dayan'97; Levine'18, …]

$$f(\boldsymbol{x}, \boldsymbol{y}) \coloneqq Q^{\theta^t}(\boldsymbol{x}, \boldsymbol{y}) \qquad \alpha = \beta = \tau \; (> 0)$$

$$q(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x}) \mu^{\theta^t}(\boldsymbol{x}) \qquad p_\theta(\boldsymbol{x}, \boldsymbol{y}) = p_\theta(\boldsymbol{y}|\boldsymbol{x}) \mu^{\theta^t}(\boldsymbol{x})$$

$$\min_{q,\theta} \; -\tau H(q) - \tau \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[Q^{\theta^t}(\boldsymbol{x}, \boldsymbol{y}) \right]$$
$$\geq -\log \mathbb{E}_{\mu^{\theta^t}(\boldsymbol{x}) p_\theta(\boldsymbol{y}|\boldsymbol{x})} \left[p(o = 1 \mid \boldsymbol{x}, \boldsymbol{y}) \right]$$

Negative variational lower bound

Define random variable $o \in \{0,1\}$, $p(o = 1) \propto exp\{ Q^{\theta^t}(x, y)/\tau \}$ (reward excitement fuc.)

SE with reward experience II -- Policy gradient

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q\left[\log p_\theta(\boldsymbol{x},\boldsymbol{y})\right] - \mathbb{E}_q\left[\; f(\boldsymbol{x},\boldsymbol{y}) \quad\right]$$

- Policy gradient

$$f(\boldsymbol{x},\boldsymbol{y}) := \log Q^{\theta^t}(\boldsymbol{x},\boldsymbol{y}) \qquad \alpha = \beta = 1$$

$$q(\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x})\mu^{\theta^t}(\boldsymbol{x}) \qquad p_\theta(\boldsymbol{x},\boldsymbol{y}) = p_\theta(\boldsymbol{y}|\boldsymbol{x})\mu^{\theta^t}(\boldsymbol{x})$$

- E-step $\quad q(\boldsymbol{y}|\boldsymbol{x}) = p_{\theta^t}(\boldsymbol{y}|\boldsymbol{x})Q^{\theta^t}(\boldsymbol{x},\boldsymbol{y}) \,/\, Z$
- M-step

$$\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}[\,\nabla_\theta \log p_\theta(\boldsymbol{y}|\boldsymbol{x})\,] = 1/Z \cdot \mathbb{E}_{\mu^{\theta^t}(\boldsymbol{x})\, p_\theta(\boldsymbol{y}|\boldsymbol{x})}\left[Q^{\theta^t}(\boldsymbol{x},\boldsymbol{y})\, \nabla_\theta \log p_\theta(\boldsymbol{y}|\boldsymbol{x})\,\right]$$

(Importance sampling est.)

$$= 1/Z \boxed{\cdot\, \nabla_\theta \mathbb{E}_{\mu^{\theta^t}(\boldsymbol{x})\, p_\theta(\boldsymbol{y}|\boldsymbol{x})}\left[Q^{\theta^t}(\boldsymbol{x},\boldsymbol{y})\,\right]}$$

(Log-derivative trick)

Conventional policy gradient

SE with adversarial experiences and other divergences -- variations of GAN

$$\min_{q,\theta} -\alpha \mathbb{H}(q) + \beta \mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[f(\boldsymbol{x}) \Big]$$

- Recall in MLE, f is a fixed function

$$f := f(\boldsymbol{x}\,;\,\mathcal{D}) = \log \mathbb{E}_{\boldsymbol{x}^*\sim\mathcal{D}}\left[\mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x}) \right]$$

- Intuitively, see f as a similarity metric that measures similarity of sample $\boldsymbol{x}$ against real data $\mathcal{D}$
- Instead of the above manually fixed metric, can we **learn** a metric f_ϕ?

SE with adversarial experiences and other divergences -- variations of GAN

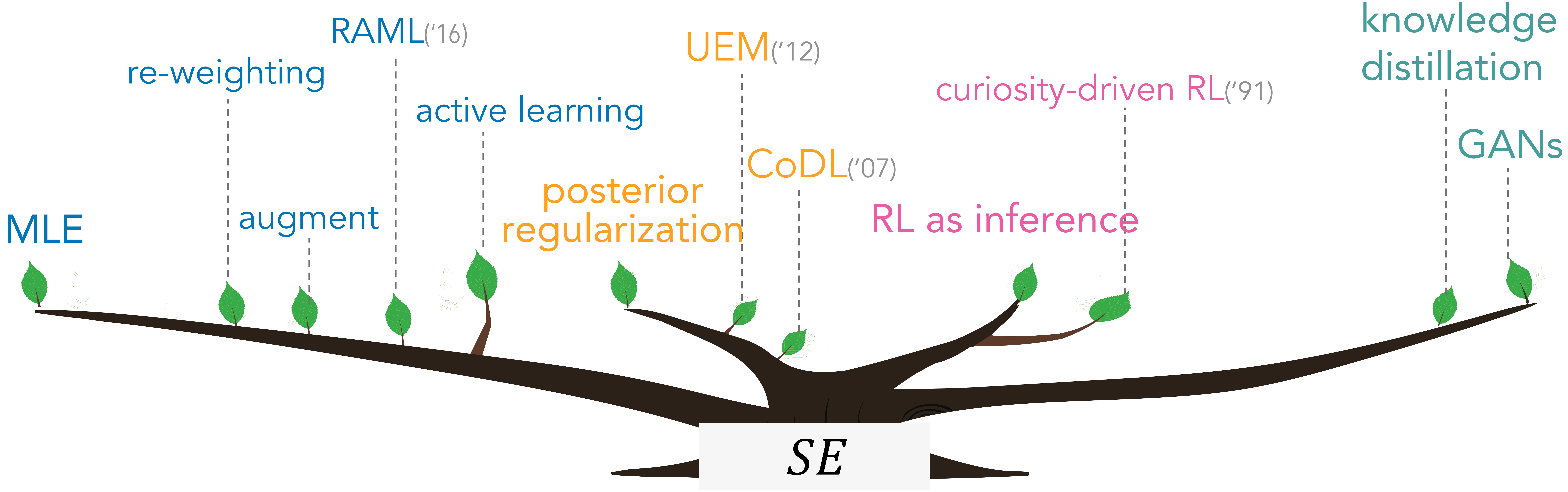
- Augment the standard objective to account for ϕ:

$$\min_{\theta} \max_{\phi} \min_{q} -\alpha \mathbb{H}(q) + \beta \mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[f_\phi(\boldsymbol{x}) \Big] + \mathbb{E}_{p_d(\boldsymbol{x})}\Big[f_\phi(\boldsymbol{x}) \Big]$$

- Set $\alpha = 0, \beta = 1$. Under mild conditions, the objective recovers:
  - Vanilla GAN [Goodfellow et al., 2014], when $\mathbb{D}$ is JS-divergence and f_ϕ is a binary classifier
  - f-GAN [Nowozin et al., 2016], when $\mathbb{D}$ is f-divergence
  - W-GAN [Arjovsky et al., 2017], when $\mathbb{D}$ is Wasserstein distance and f_ϕ is a 1-Lipschitz function

More algorithms recovered by SE

- Data augmentation / re-weighting / RAML
- Unified EM (UEM) / Constraint-driven learning (CoDL)
- Curiosity-driven RL
- Knowledge distillation



A table of all algorithms

Algorithm	f	α	β	$\mathbb{D}$
Unsupervised MLE	$f(\boldsymbol{x}; \mathcal{D})$	1	1	CE
Supervised MLE	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	1	ϵ	CE
Active Learn.	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) + u(\boldsymbol{x})$	temp., > 0	ϵ	CE
Reward-augment MLE	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	ϵ	CE
PG for Seq. Gen.	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	1	CE
Posterior Reg.	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, > 0	α	CE
Unified EM	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, $\in \mathbb{R}$	1	CE
Policy Gradient (PG)	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
+ Intrinsic Reward	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
RL as inference	$Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	temp., > 0	α	CE
Vanilla GAN	binary classifier	0	1	JSD
f-GAN	discriminator	0	1	f-divg.
WGAN	1-Lipschitz discriminator	0	1	W dist.

Paradigms not (yet) covered by SE:

- Meta learning
- Lifelong learning
- …

Interesting future work to study the connections

Outline

1. A standardized ML formalism
2. Integrated problem solving
   Plugging arbitrary experiences in learning
3. Integrated algorithm innovation

High-level Ideas

- Distinct experiences are used in learning in the **same** way
- Plug arbitrary available experiences into the learning procedure!

$$f = w_1 \cdot f(\boldsymbol{x} \mid \;) + w_2 \cdot f(\boldsymbol{x} \mid \;) + w_3 \cdot f(\boldsymbol{x} \mid \;) + w_4 \cdot f(\boldsymbol{x} \mid \;) + \cdots$$

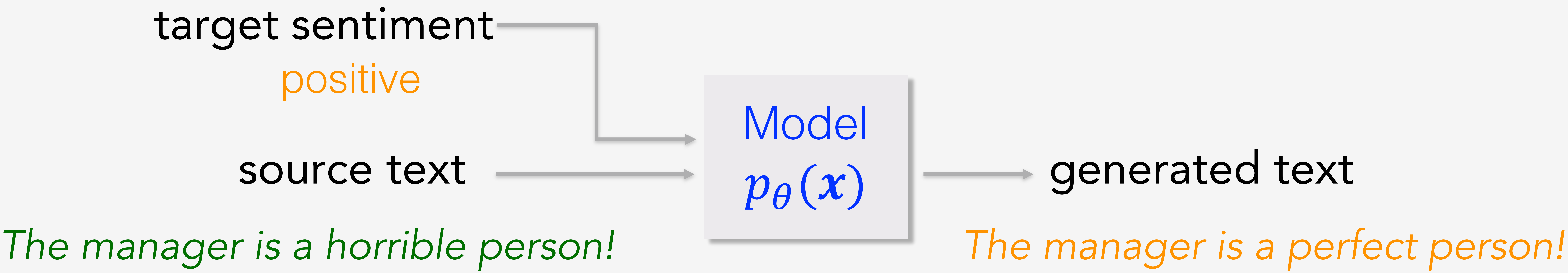
Focus on **what** to use, instead of worrying about **how** to use

Problem: Controllable Text Generation

Ex. controlling sentiment

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

Key challenge: no direct supervision data

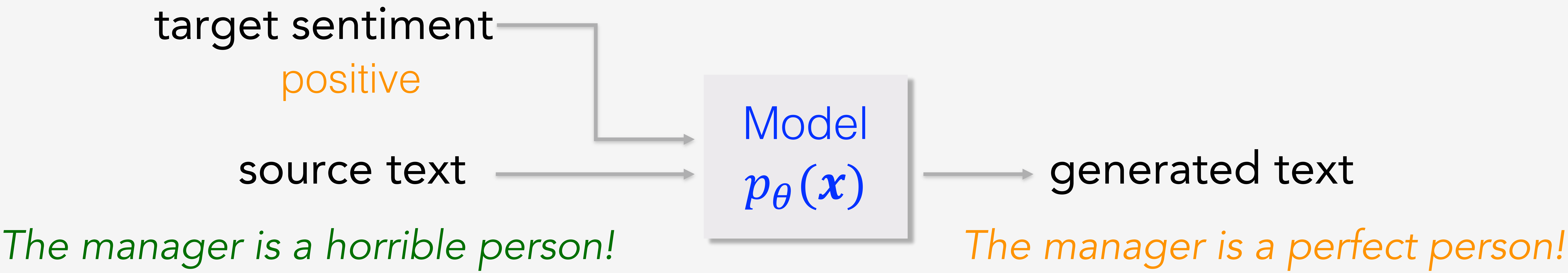


[Hu et al., ICML'17]

Designing the Solution

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

1) Consider what experiences to use



[Hu et al., ICML’17]

Designing the Solution

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

1) Consider what experiences to use

Sentiment classifier

The manager is a perfect person! → *Sentiment classifier* → *logit of target sentiment*

[Hu et al., ICML'17]

Designing the Solution

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

1) **Consider what experiences to use**

Sentiment classifier

[Hu et al., ICML'17]

Designing the Solution

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

1) Consider what experiences to use

Sentiment classifier

Self-construction data examples

target sentiment

negative

source text

The manager is a horrible person!

Model

$p_\theta(\boldsymbol{x})$

generated text

The manager is a horrible person!

[Hu et al., ICML'17]

Designing the Solution

Goals: generating a new sentence that has the target sentiment while preserving all other aspects

1) Consider what experiences to use

Sentiment classifier *Self-construction data examples* *Language model* ...

The manager is a perfect person! → *Language model* → *perplexity score*

[Hu et al., ICML'17]

Designing the Solution

1) **Consider what experiences to use**

S *Sentiment classifier*

Self-construction data examples

L *Language model*

2) **Plug experiences into the algorithm**

[Hu et al., ICML'17]

Designing the Solution

1) Consider what experiences to use

Sentiment classifier

Self-construction data examples

Language model

2) Plug experiences into the algorithm

Source text: The manager is a horrible person!

Experiences plugged in:	*Resulting generated text:*
$f = f(x\|S)$	good good good good …
$+ f(x\|\;)$	The manager is a delicious person!
$+ f(x\|L)$	The manager is a perfect person!

[Hu et al., ICML'17]

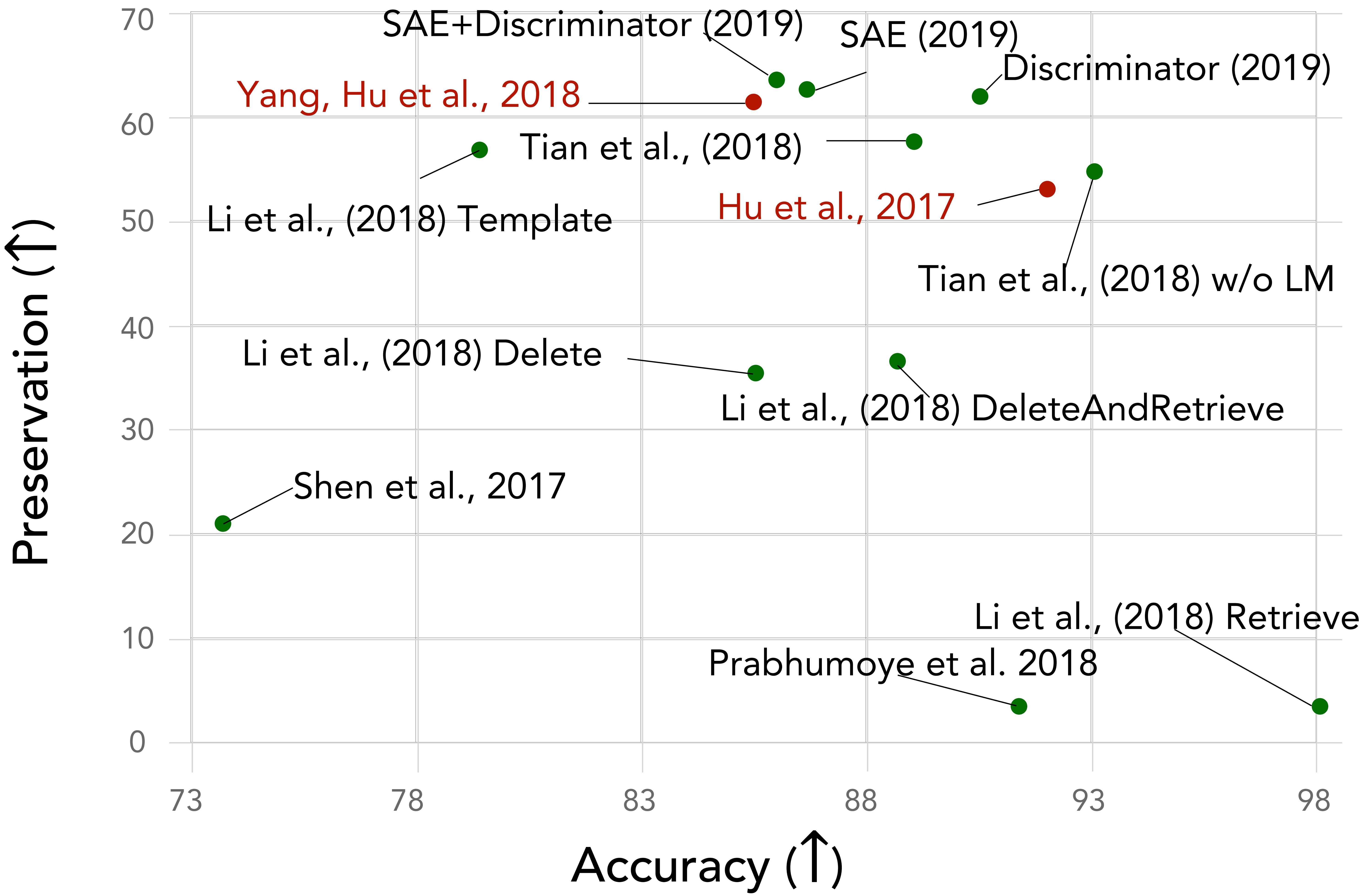
Experimental Results

	Accuracy (↑)	Perservation (↑)	Language quality (↓)
Sentiment classifier	**98.9** 😁	0.1	326.1
+ Self-construction data examples	87.7 😁	**65.6** 😁	115.6
+ Language model	91.2 😁	57.8 😁	**47.2** 😁

[Hu et al., ICML’17]

Controllable text generation as an NLP benchmark

[Tikhonov et al., EMNLP2019, Style Transfer for Texts: Retrain, Report Errors, Compare with Rewrites]



Controllable text generation in various applications

follow-up work

- Emotional chatbot [e.g. Rashkin et al., 2018; Zhou et al., 2018]
- Neutralizing bias in text [e.g. Chen et al., 2018; Youngmann, 2019]
- Generating text adversarial examples [e.g. Zhao et al., 2018]
- Data augmentation [e.g. Verma et al., 2018; Malandrakis et al., 2019]

Controllable Text Generation

Other work

Rewriting content

He scored 23 points and pulled down 8 rebounds .

Name	LeBron James
Points	32
Rebounds	4
Assists	7

LeBron James scored 32 points, pulled down 4 rebounds, and added 7 assists.

[Lin et al., 2019]

Guiding conversation flow

Hi, how are you today?

Fine. Just finished riding along the river.

Cool! You can ride bikes, listen to music there too.

Yes. I like Taylor Swift.

I love to sing her songs! Do you?

Not really. I cannot sing well.

How about dancing? I love dancing!

[Tang et al., ACL'19]

Open-source toolkits for composing ML solutions

Apps	Text Generation	Text Analysis	Medical Diagnosis	…
High-level ML toolkits	**Texar** *[Hu et al., ACL'19]*	**Forte** *[Liu et al., EMNLP'20]*		
Base ML platforms	TensorFlow	PyTorch	…	

https://github.com/asyml
327 2,017
85 603

We welcome any contributions !

Tutorials & invited talks:

AAAI-20, New York
NVIDIA GTC GPU TECHNOLOGY CONFERENCE
ICML | 2019
KDD2020
ICERM
ODSC

Take Aways

- Designing learning solutions by plugging arbitrary experiences into the integrated algorithm
- Rich problems and results in controllable text generation

Outline

1 Unifying formulation of extensive algorithms

2 Integrated problem solving

3 Integrated algorithm innovation

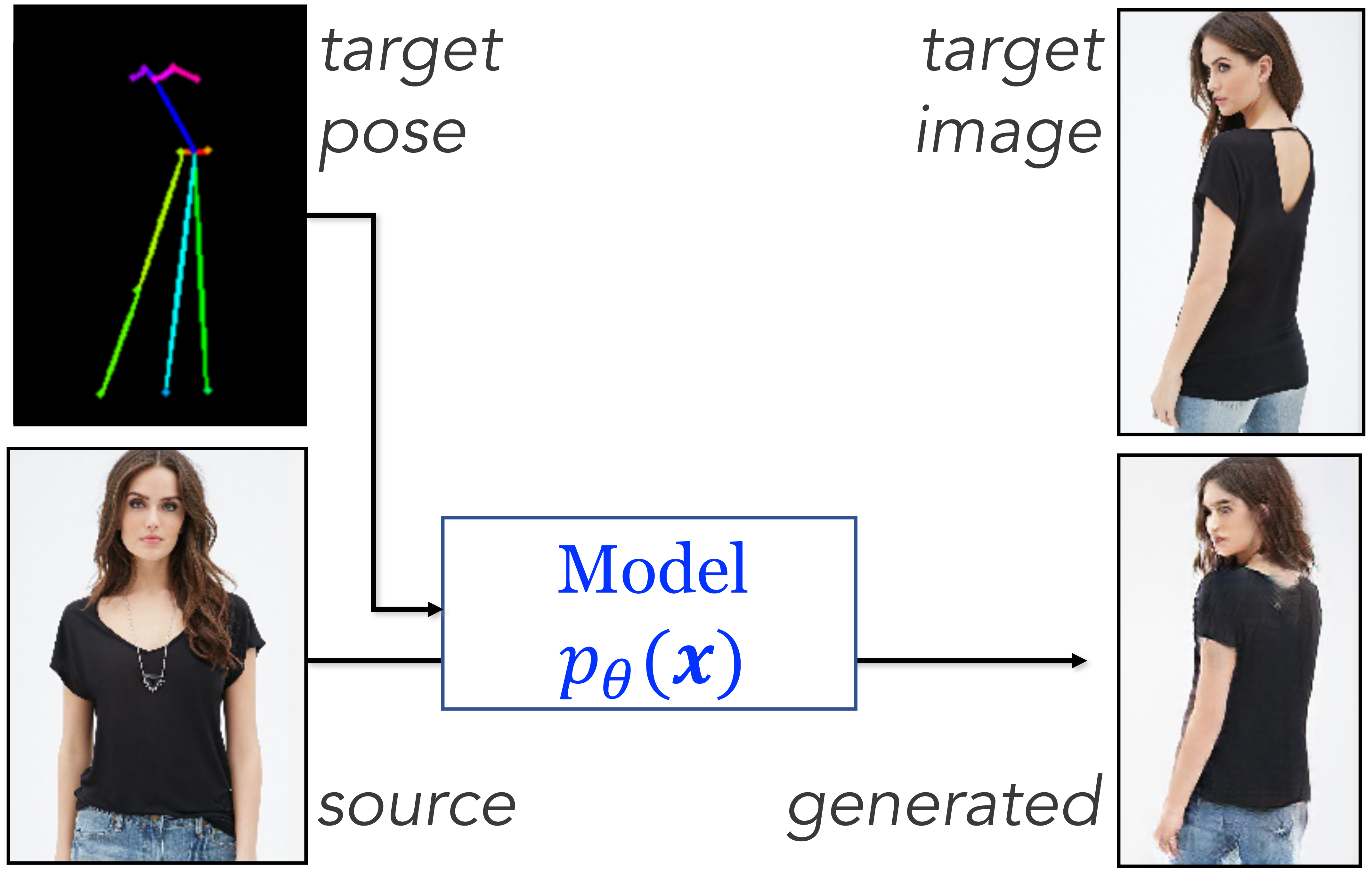
An advance in one area unlocks advances in many others

High-level Ideas

- Unifying perspective of extensive algorithms

- Systematic idea transfer and solution exchange
  - Accelerate innovations across areas
  - Solving challenges in one area by applying well-known solutions in another

Problem: Fashion Image Generation

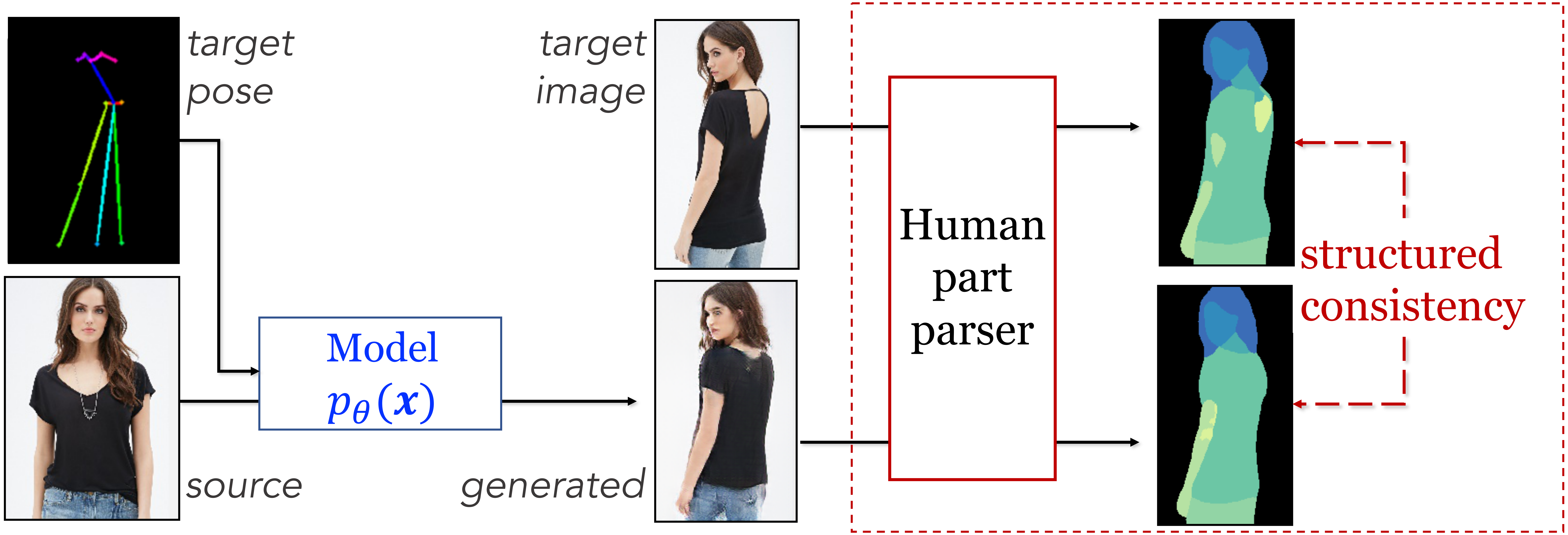


[Hu et al., NeurIPS’18]

Problem: Fashion Image Generation

Human body constraint:

$f\ (\boldsymbol{x}\ |\ \ )$ =match score



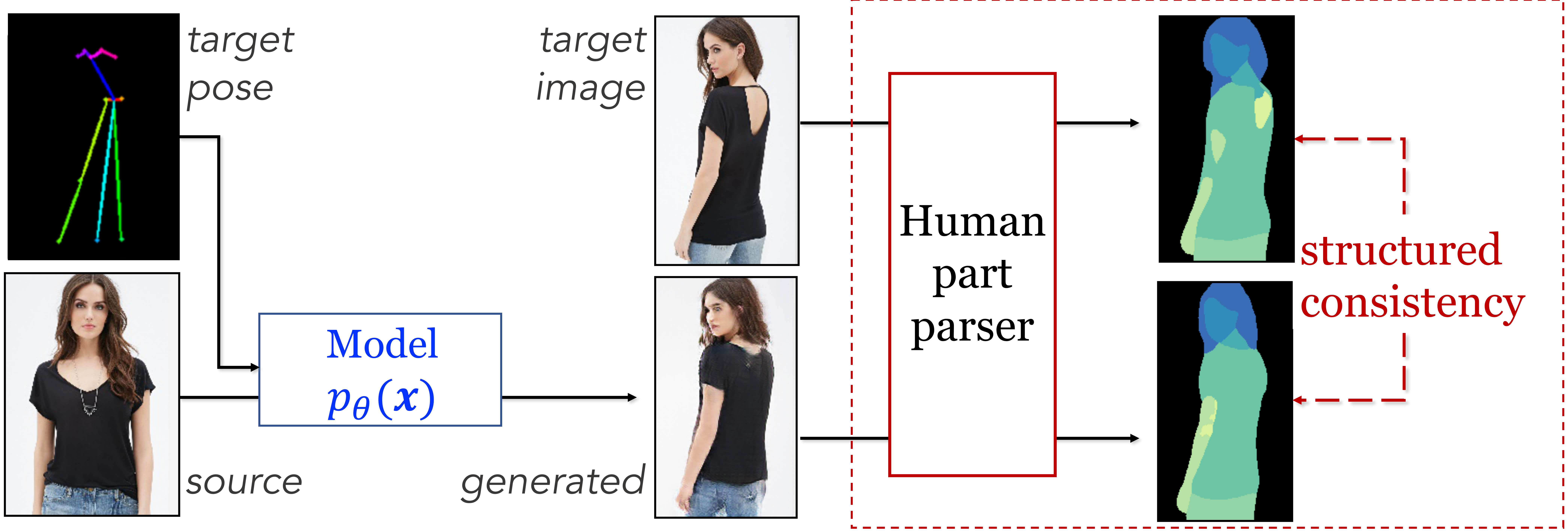
[Hu et al., NeurIPS'18]

Problem: Fashion Image Generation

Key challenge: the constraint is not accurate enough

Human body constraint:

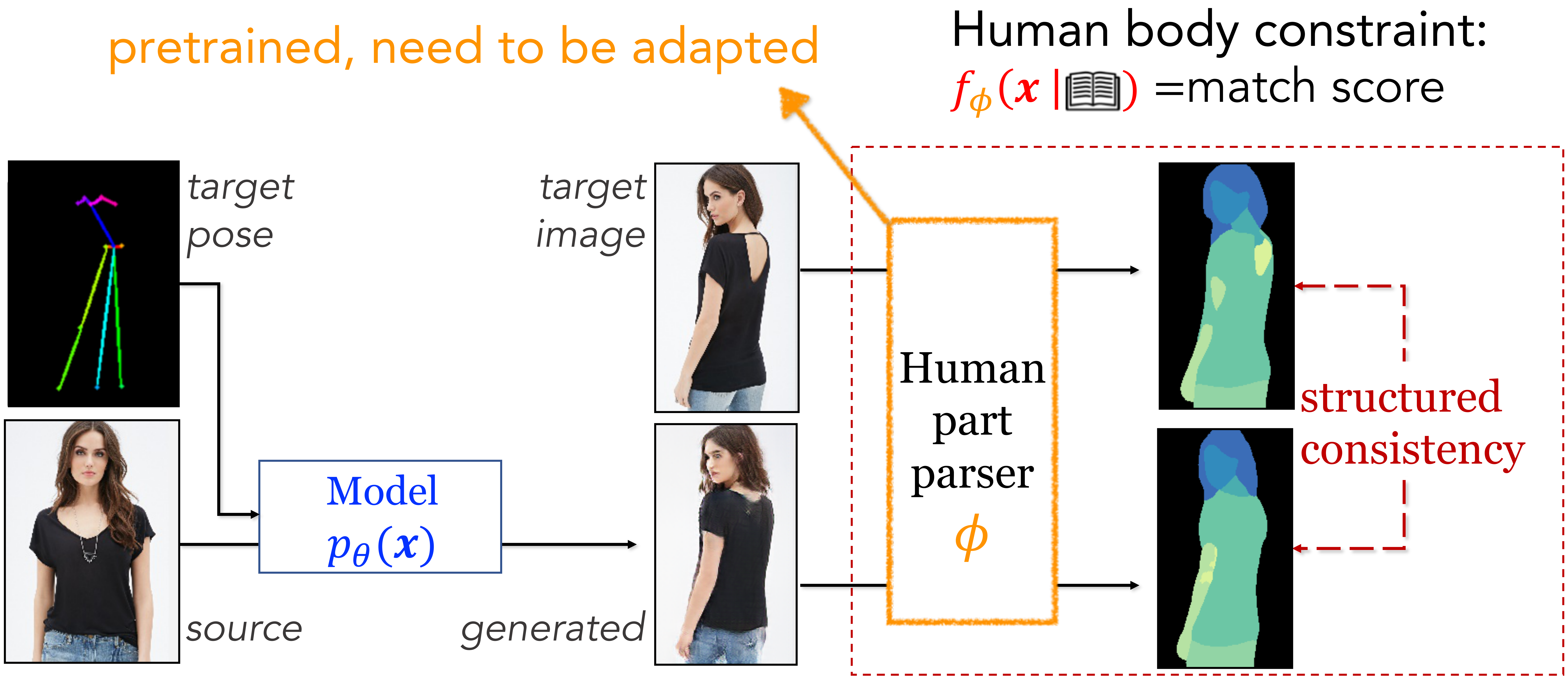
$f\ (\boldsymbol{x}\ |\ \ )$ =match score



[Hu et al., NeurIPS'18]

Problem: Fashion Image Generation

Key challenge: the constraint is not accurate enough



[Hu et al., NeurIPS'18]

Solving the challenge under the unifying perspective

Learning problem: adapting the constraint while learning the model

$f_{\phi}(\boldsymbol{x}|\text{📖})$

1 auxiliary in closed form $q_{\phi}(\boldsymbol{x})$

2 update the model $p_{\theta}(\boldsymbol{x})$

3

How to adapt the constraint?

Solving the challenge under the unifying perspective

Learning problem: adapting the constraint while learning the model

$f_{\phi}(\boldsymbol{x}|\ )$

1. auxiliary in closed form $q_{\phi}(\boldsymbol{x})$
2. update the model $p_{\theta}(\boldsymbol{x})$

3.

?

How to adapt the constraint?

Import a solution from another area!

constraint  reward

adapting constraint ⟵ adapting reward

MaxEnt inverse RL [Ziebart'08]

adapting reward by:

$$\min_{\phi} -\mathbb{E}_{\boldsymbol{x}^*}\left[\log q_{\phi}(\boldsymbol{x}^*)\right]$$

[Hu et al., NeurIPS'18]

Solving the challenge under the unifying perspective

Learning problem: adapting the constraint while learning the model

$f_{\phi}(\boldsymbol{x}|\;)$

1. auxiliary in closed form $q_{\phi}(\boldsymbol{x})$
2. update the model $p_{\theta}(\boldsymbol{x})$

3. adapt constraint by:

$$\min_{\phi} -\mathbb{E}_{\boldsymbol{x}^{*}\sim}\left[\log q_{\phi}(\boldsymbol{x}^{*})\right]$$

Import a solution from another area!

constraint  reward

adapting constraint ⟵ adapting reward

MaxEnt inverse RL [Ziebart'08]

[Hu et al., NeurIPS'18]

Solving the challenge under the unifying perspective

Learning problem: adapting the constraint while learning the model

$f_\phi(\boldsymbol{x}|\ )$

1. auxiliary in closed form $q_\phi(\boldsymbol{x})$
2. update the model $p_\theta(\boldsymbol{x})$
3. adapt constraint by:

$$\min_\phi -\mathbb{E}_{\boldsymbol{x}^*\sim}\left[\log q_\phi(\boldsymbol{x}^*)\right]$$

[Hu et al., NeurIPS'18]

A closer look:

*Resemble **adversarial learning**, but sampling from q_ϕ instead of p_θ*

Bonus: This is more effective!

[Wu et al., 2020]

$$\mathbb{E}_{\boldsymbol{x}^*\sim}\left[\nabla_\phi f_\phi(\boldsymbol{x}^*|\ )\right] - \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\nabla_\phi f_\phi(\boldsymbol{x}|\ )\right]$$

take gradient w.r.t ϕ

$$q_\phi(\boldsymbol{x}) = \exp\left\{\frac{\alpha \log p_\theta(\boldsymbol{x}) + f_\phi(\boldsymbol{x}\ |\ )}{\alpha + \beta}\right\} / Z$$

Experimental Setup

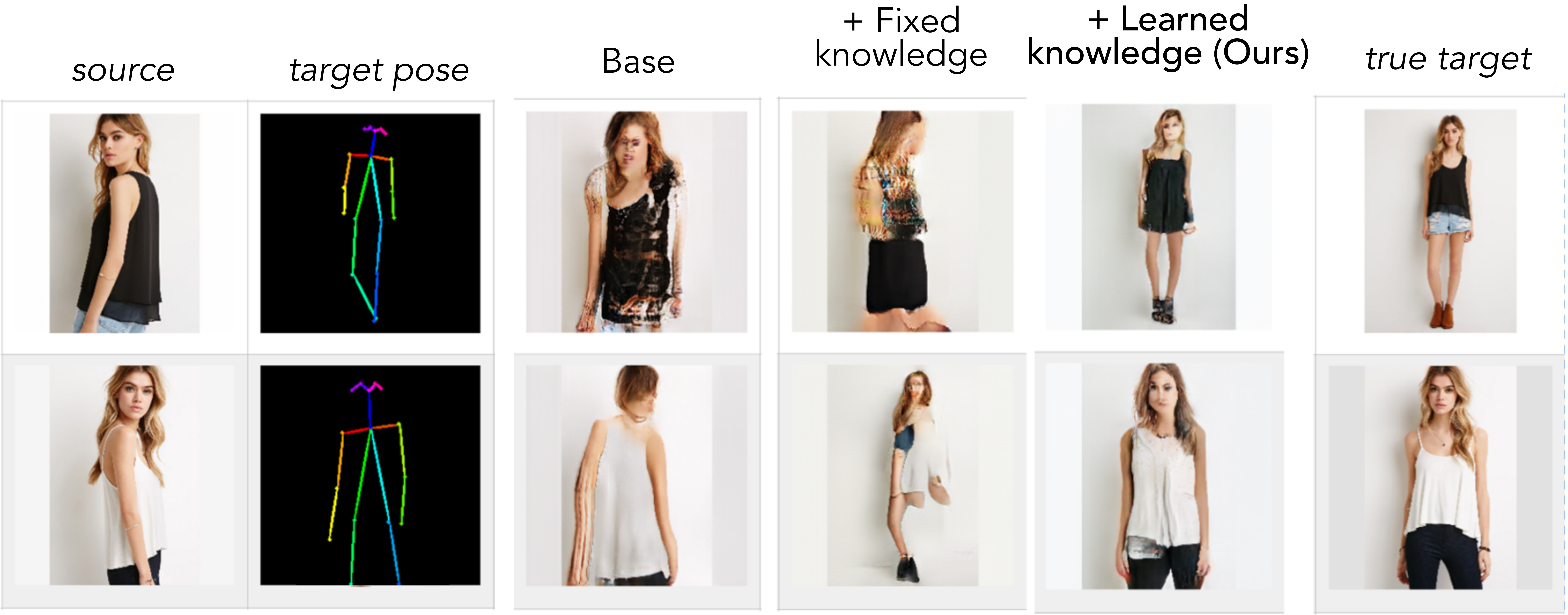
- **Dataset:** DeepFashion [Liu et al'16]
  - 81K data examples
  - Image size: 256x256

- **Comparison methods:**
  - Base: GANs + supervised
  - + Fixed knowledge
  - + Learned knowledge *(Ours)*
  - Previous work [Isola et al.'17, Ma et al.'17]

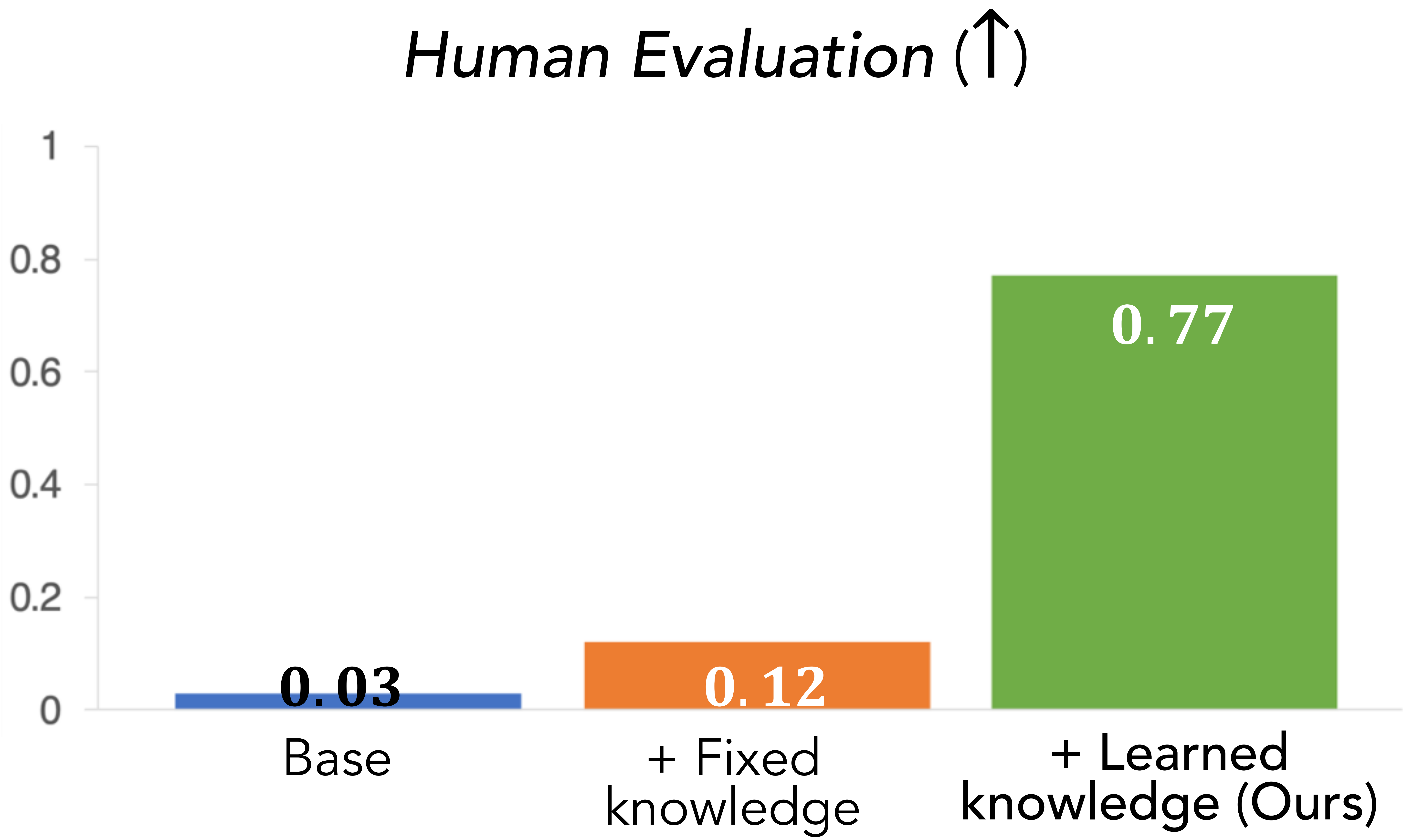
Not exploiting any structured knowledge of the problem

[Hu et al., NeurIPS'18]

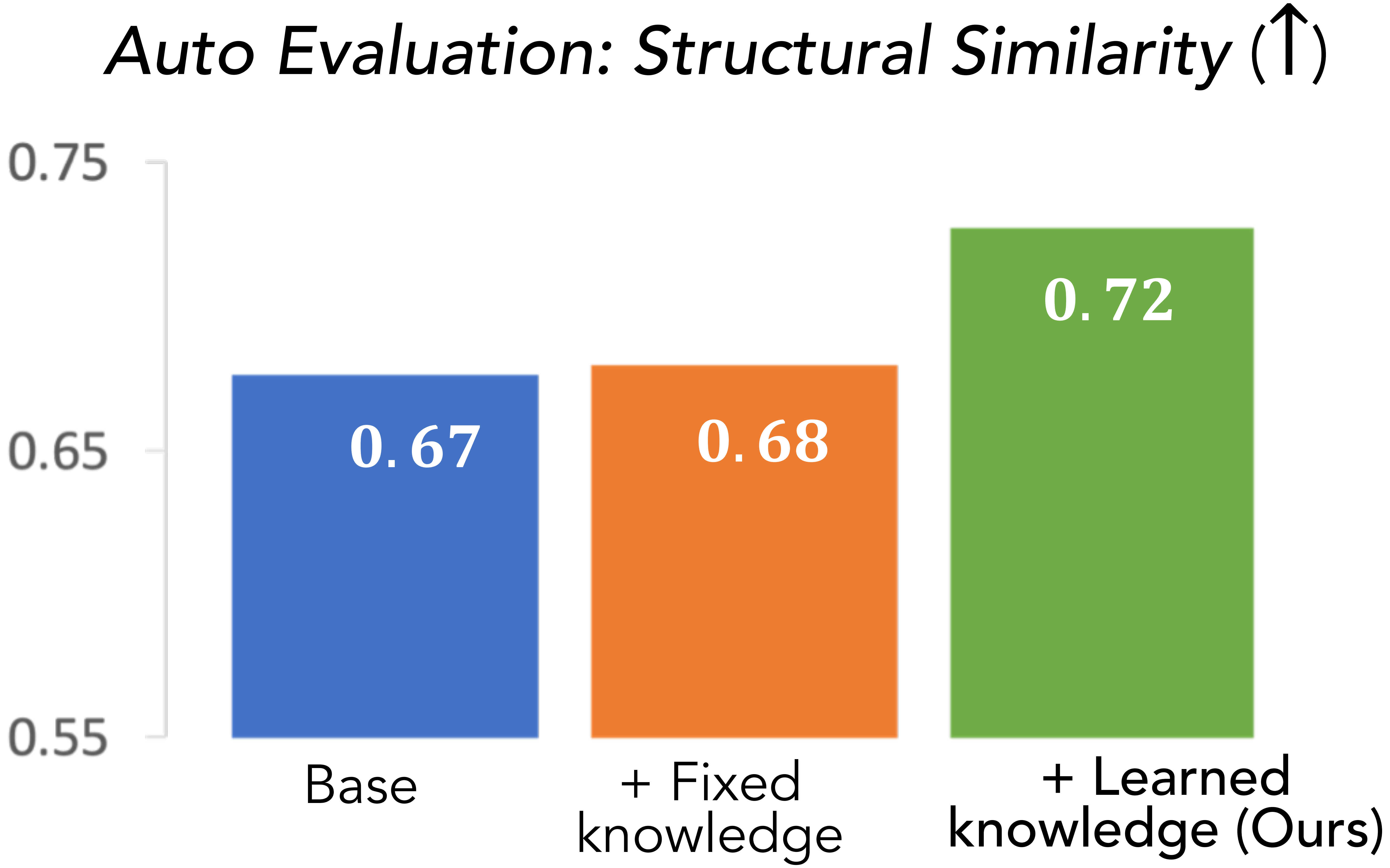
Experimental Results



Experimental Results



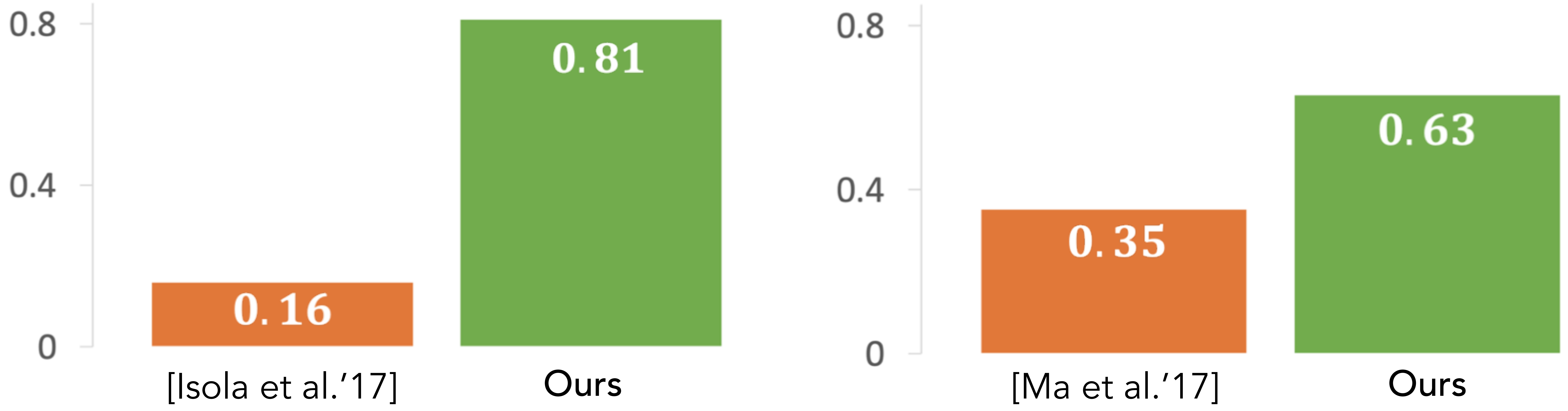
Experimental Results



Experimental Results

Pairwise Comparisons with Previous Work

Human Evaluation (↑)



Integrated innovation across areas

Other work

Off-the-shelf reward learning algorithms

- MaxEnt inverse RL [Ziebart'08]
- Intrinsic reward learning [Pathak'17, Zheng'18, …]

Constraint adapting [Hu et al., NeurIPS'18]

Automated data augmentation & re-weighting [Hu, Tan, et al., NeurIPS'19]

Stablized policy training

- Proximal Policy Optimization [Schulman'2017]

Stablized GAN training [Wu et al, 2020]

Importance weighted VAE [Burda'15]

Importance weighted GANs [Hu et al., ICLR'18]

Integrated innovation across areas

Other work

Stablized policy training → Stablized GAN training [Wu et al, 2020]

- Proximal Policy Optimization [Schulman'2017]

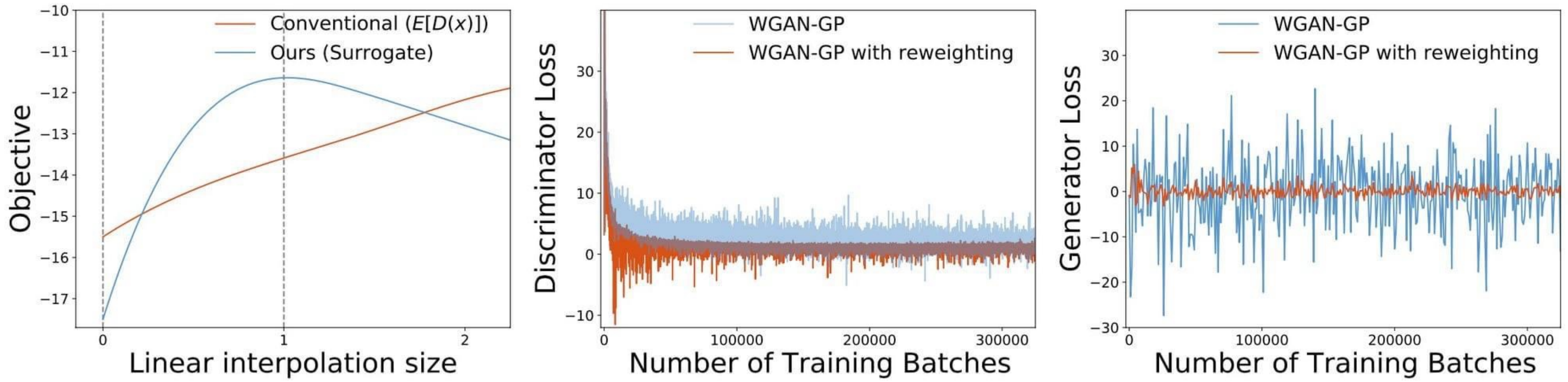


Figure 1: Illustration of the proposed approach for stabilizing GAN training. Results are from the CIFAR-10 experiment in Sec. 4.1. **Left:** The conventional and surrogate objectives for generator training, as we interpolate between the initial generator parameters $\boldsymbol{\theta}_{old}$ and the updated generator parameters $\boldsymbol{\theta}_{new}$ which we compute after one iteration of training. The updated $\boldsymbol{\theta}_{new}$ obtains maximal surrogate objective. The surrogate objective imposes a penalty for having too large of a generator update, since the curve starts decreasing after $x = 1$. In contrast, the conventional objective (for WGAN-GP) keeps increasing with larger generator updates. **Middle and right:** Discriminator and generator losses w/ and w/o sample re-weighting. WGAN-GP with our re-weighting plugged in shows lower variance in both discriminator and generator losses throughout training (and achieves better final performance as shown in Sec. 4.1).

Method	**IS** (↑)	**FID** (↓)
Real data	11.24±.12	7.8
WGAN-GP (2017)	7.86±.08	-
CT-GAN (2018)	8.12±.12	-
SN-GANs (2018)	8.22±.05	21.7±.21
WGAN-ALP (2020)	8.34±.06	12.96±.35
SRNGAN (2020)	8.53 ±.04	19.83
AutoGAN (2019)	8.55±.10	12.42
Ours (re-weighting only)	8.45±.14	13.21±.60
Ours (full)	**8.69±.13**	**10.70±.10**

Table 1: CIFAR-10 results. Our method is run 3 times for average and standard deviation.

Length	MLE	SeqGAN [53]	LeakGAN [19]	RelGAN [36]	WGAN-GP [18]	Ours	Real
20	9.038	8.736	7.038	6.680	6.89	**5.67**	5.750
40	10.411	10.310	7.191	6.765	6.78	**6.14**	4.071

Table 2: Oracle negative log-likelihood scores (↓) on synthetic data.

Method	**BLEU-2** (↑)	**BLEU-3** (↑)	**BLEU-4** (↑)	**BLEU-5** (↑)	$\mathbf{NLL}_{gen}$ (↓)
MLE	0.768	0.473	0.240	0.126	2.382
SeqGAN [53]	0.777	0.491	0.261	0.138	2.773
LeakGAN [19]	0.826	0.645	0.437	0.272	2.356
RelGAN (100) [36]	0.881	**0.705**	**0.501**	0.319	2.482
RelGAN (1000) [36]	0.837	0.654	0.435	0.265	2.285
WGAN-GP [18]	0.872	0.636	0.379	0.220	**2.209**
Ours	**0.905**	0.692	0.470	**0.322**	2.265

Table 3: Results on EMNLP2017 WMT News. BLEU measures text quality and NLL_{gen} evaluates sample diversity. We copied the results of previous text GAN models from [36], where RelGAN (100) and RelGAN (1000) use different hyperparameters as reported in the paper.

Take Aways

- Systematic idea transfer and solution exchange
  - Accelerate innovations across areas
  - Solutions to new problems come for free !

Recap

1 A standardized ML formalism

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \alpha\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \beta\mathbb{H}(q)$$

2 Integrated problem solving

Plugging arbitrary experiences in learning

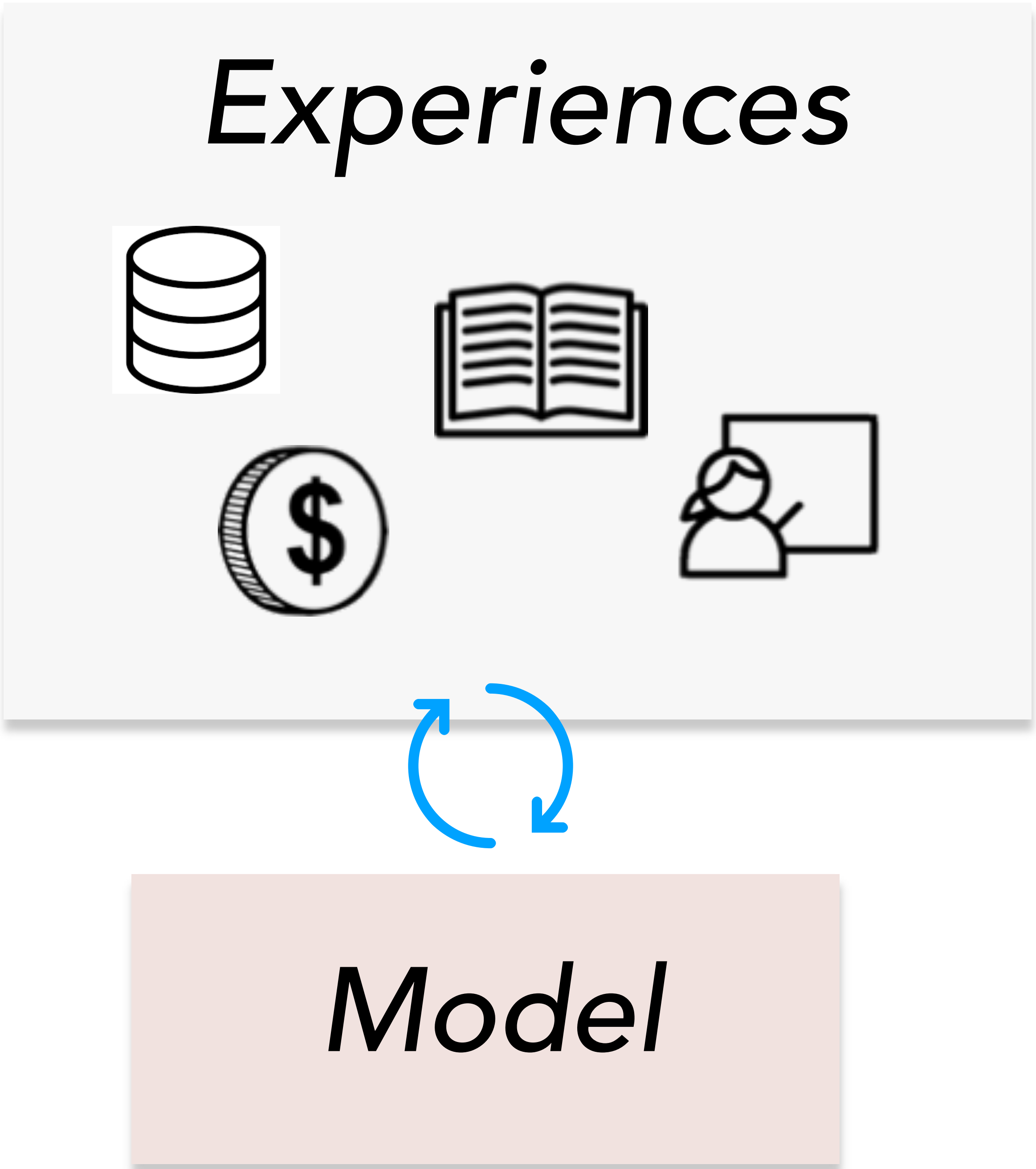
3 Integrated algorithm innovation

An advance in one area unlocks advances in many others

Expanding the unifying understanding

- **How can we connect the dots of the wider landscape of ML?**
  Ex. learning in changing, interactive environment (online learning, lifelong learning, etc)

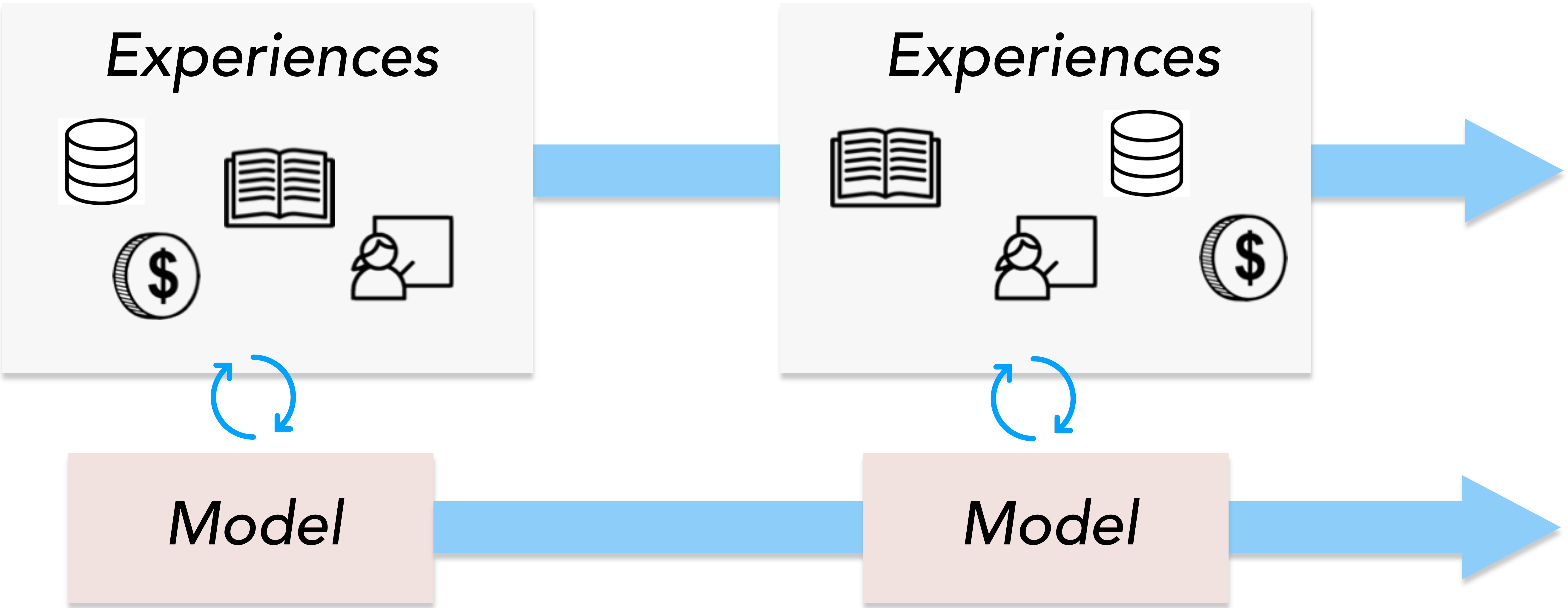
Learning from multiple experiences



Expanding the unifying understanding

- **How can we connect the dots of the wider landscape of ML?**
  Ex. learning in changing, interactive environment (online learning, lifelong learning, etc)

Learning from multiple, **evolving, interactive** experiences



Expanding the unifying understanding

- How can we connect the dots of the wider landscape of ML?

- How can we provide guarantees for learning with multiple experiences?
  Ex. get better performance every time adding new (possibly noisy)

From standardization to automation

- Unified, standardized algorithmic representation

↓

- Automated manipulation and creation of learning solutions

  enriched "AutoML" capability (beyond hyperparameter & arch search)

↓

- Improving ML accessibility to broader users

- Healthcare
- Energy eng.
- Bio eng.

…

Domain knowledge in various forms

Predictions

- Learning from all experiences
- Automated ML engine

- HCI
- PL
- System

…

Food for Thought:
How Far Would This Take Us?

- Physics

It is only slightly overstating the case to say that ***physics is the study of symmetry****.*

-- Phil Anderson (1923-2020), Physicist, Nobel laureate

Food for Thought:
How Far Would This Take Us?

- ## Physics

Maxwell's equations — 1861

General relativity — 1910s

Standard model — 1970s

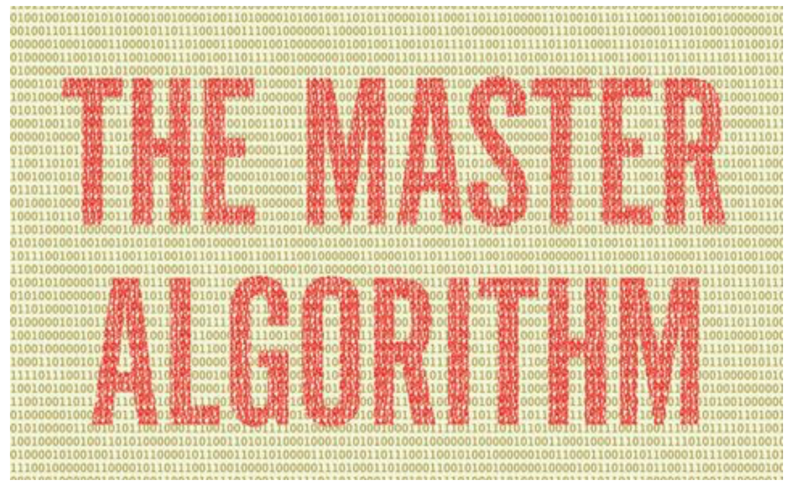
? Theory of everything

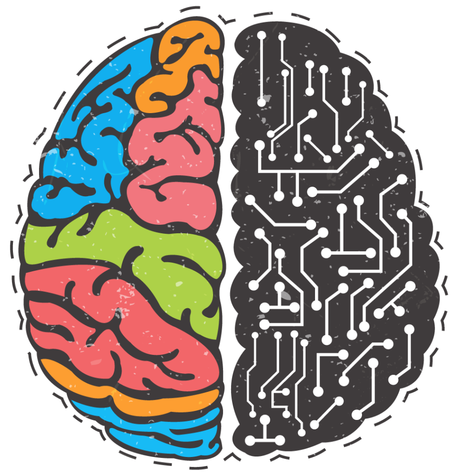
- ## Machine Learning

Unified way of thinking

- Systematic understanding
- Automated solution creation
- Improved ML accessibility

?





Thanks !