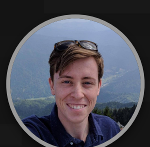
Sampling from Discrete Energy-Based Models with Quality/Efficiency Trade-Offs

ML Collective's Deep Learning Classics and Trends - April 22nd 2022




Bryan Eikema
Germán Kruszewski
Hady Elsahar
Marc Dymetman

NAVER LABS Europe

The arrival of Large Pre-trained Language Models

Large Pre-Trained Language Models are becoming **ubiquitous** thanks to their **strong generalization capabilities**.

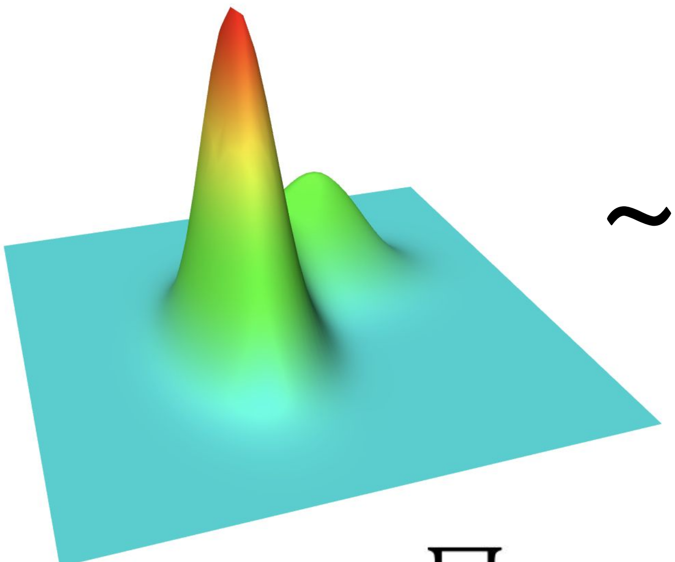
GPT-2 GPT-3 GitHub Copilot HyperCLOVA

Yet, they can generate text suffering from many problems, among which:

- ⚖️ **Bias:** Unfair misrepresentation of a person or group.
- 🤬 **Toxicity:** Offensive content.
- 🙃 **Incoherence:** Lack of consistency and continuity between fragments.
- 👻 **Hallucinations:** Unfounded content.

Generation from Pre-trained Language Models

HyperCLOVA



$\sim$

"English keeps update of Manchester Manchester Arena after Manchester City 0-0 Arsenal 100 5 19 We have brought in a leading expert to declare someone Sawaya defender would feel was amazing even if brought in."

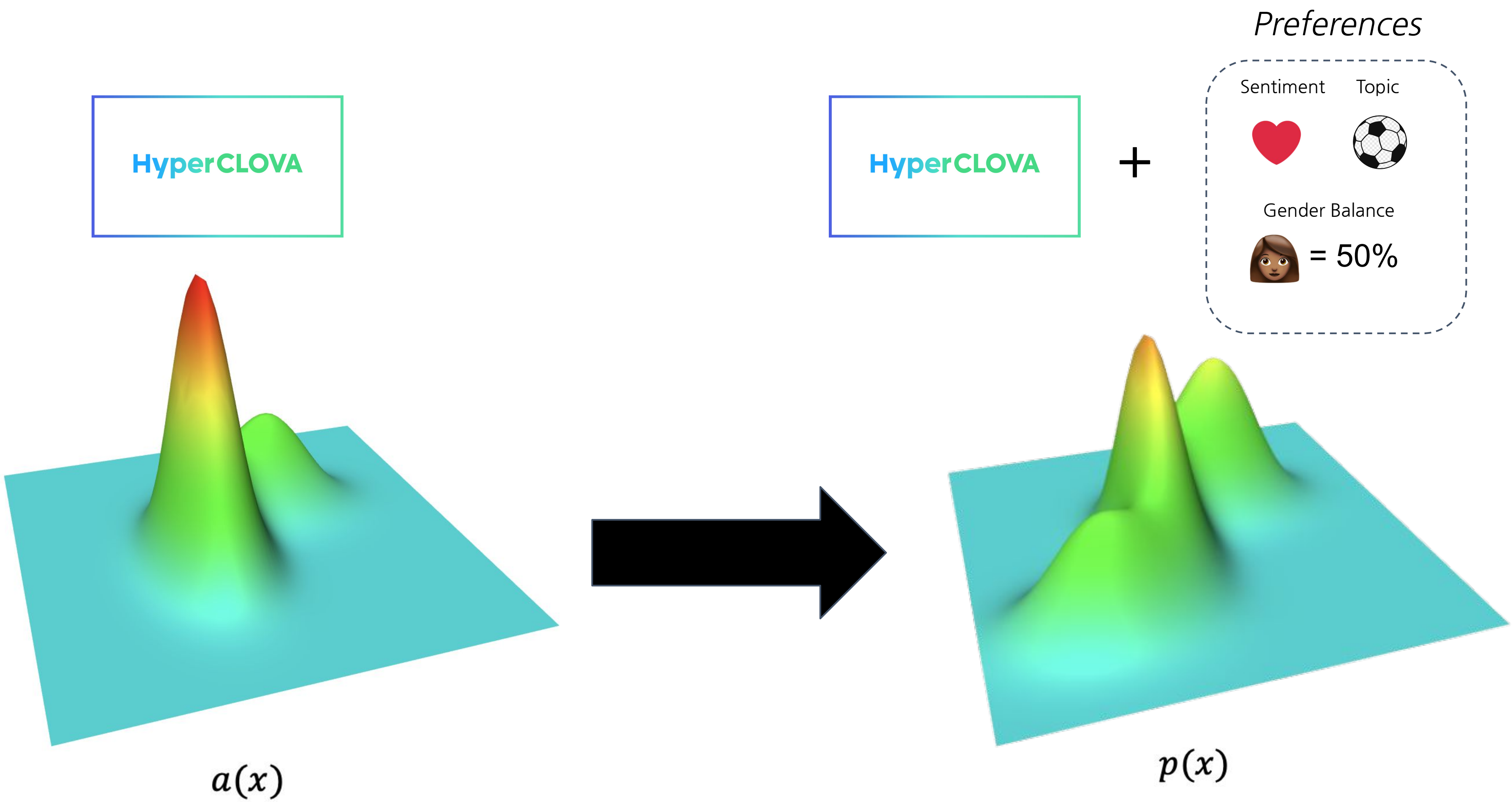
"And what about everyone your boss wants you to love? Well it can be amazing to read about you March Madness-loved heroics, laysh, charisma, politics, partying, etc. What"

"At 13 months old your brain becomes just the perfect tool for a multi cutting edge nano-printer way to recruit and control data filled with amazing 3D adaptive lighting, support for controllers such as FT"

$$a(x) = \prod_t a(x_t | x_0 \ldots x_{t-1})$$

Pre-trained Language Model distribution

Controlling Pre-trained Language Models



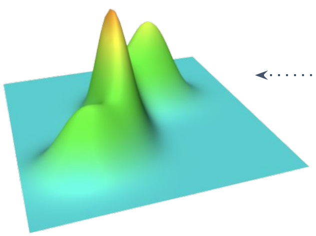
Pre-trained Language Model distribution

Controlled Language Model distribution

GDC Generation with Distributional Control:

A Two-steps Framework For Controlled Language Generation

Khalifa, Muhammad, Hady Elsahar, and Marc Dymetman. "A distributional approach to controlled text generation." In Proceedings of *ICLR* (2021).



$p(x)$

The target distribution is the probability mass function that:

(1) satisfies the desired preferences, including

🔥 To the best of our knowledge, our approach is the **only method available** to handle such preferences .

+ve Sentiment = 100% ❤️

Pointwise preferences represent properties of *individual sequences*.

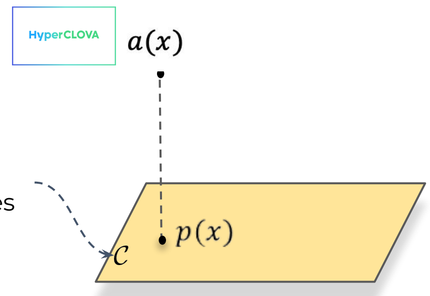
Gender Balance

👩🏾 = 50%

Distributional preferences represent properties of the *full distribution*.

(2) avoids "catastrophic forgetting":
Minimally deviates from the original Language Model.

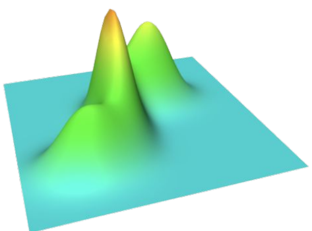
The set of all distributions that satisfy the preferences



GDC Generation with Distributional Control:

Step 1: Define the EBM

Khalifa, Muhammad, Hady Elsahar, and Marc Dymetman. "A distributional approach to controlled text generation." In Proceedings of *ICLR* (2021).



$$p(x) = \frac{P(x)}{\sum_x P(x)}$$

- **Pointwise preferences**

$$P(x) \doteq a(x)b(x)$$

Sentiment classifier

- **Distributional preferences**

Topic classifier

Gender classifier

$$\mathbb{E}_{x\sim p}\phi_1(x) = \bar{\mu}_1 \quad \text{100\%}$$

$$\mathbb{E}_{x\sim p}\phi_2(x) = \bar{\mu}_2 \quad \text{50\%}$$

$$\dots$$

$$\mathbb{E}_{x\sim p}\phi_n(x) = \bar{\mu}_n$$

$$P(x) \doteq a(x)e^{\lambda\cdot\phi(x)}$$

Moment Matching

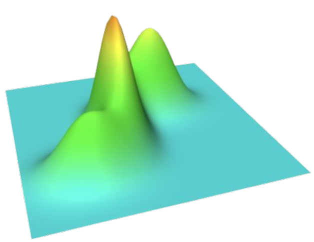
ℹ️ We say P(x) is an "Energy Based Model" or EBM.

GDC Generation with Distributional Control:

Step 2: Fine-tune a LM to approximate p using the DPG algorithm

Khalifa, Muhammad, Hady Elsahar, and Marc Dymetman. "A distributional approach to controlled text generation." In Proceedings of *ICLR* (2021).

Converts the EBM $P(x)$ into an autoregressive model π_θ which minimizes $CE(p, \pi_\theta)$

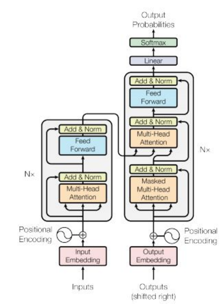


$$P(x) \doteq a(x) e^{\lambda \cdot \phi(x)}$$

$$p(x) = \frac{P(x)}{\sum_x P(x)}$$

$$\nabla_\theta CE(p, \pi_\theta) = -\mathbb{E}_{x \sim q} \frac{p(x)}{q(x)} \nabla_\theta \log \pi_\theta(x)$$

Fine tuning the Language Model to minimize cross entropy with the EBM distribution.



π_θ

How can we sample from the EBM?

For sake of example, let's define the EBM P as

$$P(x) \doteq a(x)\mathbb{I}[\text{“wikileaks”} \in x]$$

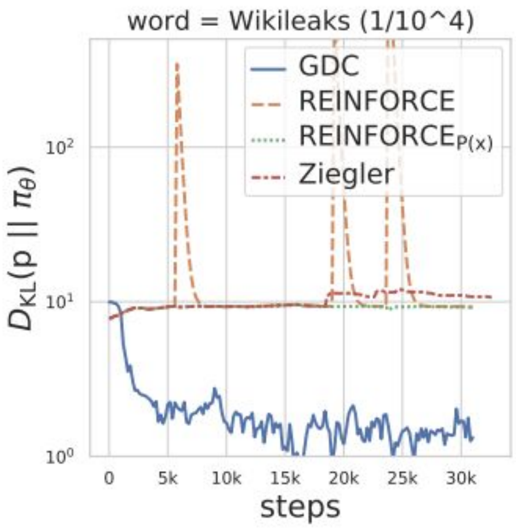
GPT-2 small

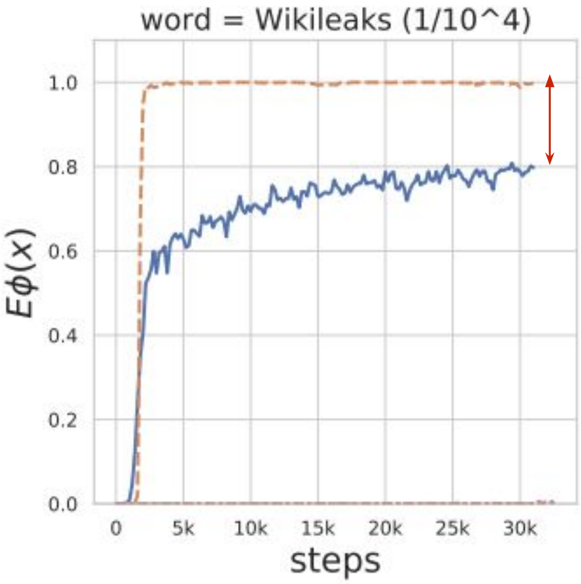
$$= \begin{cases} a(x) & \text{“wikileaks”} \in x \\ 0 & \text{otherwise} \end{cases}$$

GDC Generation with Distributional Control:

Step 2: Fine-tune a LM to approximate p using the DPG algorithm

Khalifa, Muhammad, Hady Elsahar, and Marc Dymetman. "A distributional approach to controlled text generation." In Proceedings of *ICLR* (2021).





Much better, but
quite not there!

Rejection sampling

$$P(x) \doteq a(x)\mathbb{I}[\text{“wikileaks”} \in x]$$

⚠️ β must upper-bound P(x)/q(x) over the full domain

Require: P, q, β

while True **do**

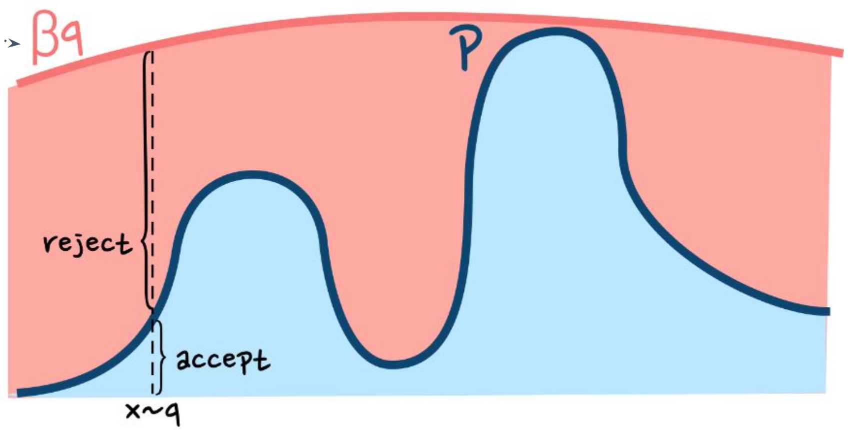
  $x \sim q$

  $r_x \leftarrow P(x)/\beta q(x)$

  $u \sim U_{[0,1]}$

  if $u \leq r_x$ **then**

    output x



$$q(x) \doteq a(x)$$

$$\frac{P(x)}{q(x)} = \frac{a(x)\mathbb{I}[\text{“wikileaks”} \in x]}{a(x)} = \mathbb{I}[\text{“wikileaks”} \in x] \leq 1$$

💡 Using the fine-tuned language model π_θ as a proposal could be much better!

Disadvantages of Rejection Sampling

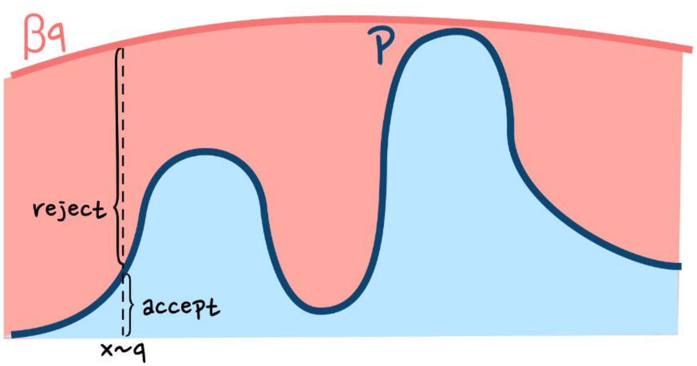
How can we find β for P(x)/q(x) when q(x)= π_θ(x) is a fine-tuned language model?

In general, we hit the following problems of RS:

👎 β may not be easy to find.

👎 β may be too large to be practical.

👎 β may not exist.

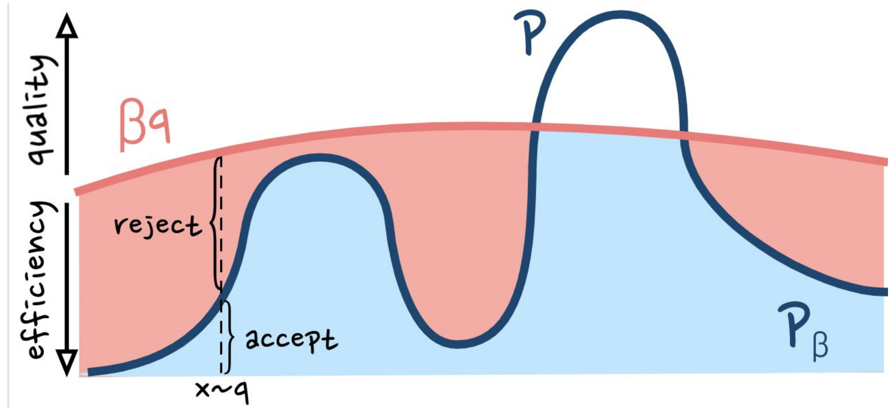


Quasi Rejection Sampling (QRS)

Bryan Eikema, Germán Kruszewski, Hady Elsahar, and Marc Dymetman,. "Sampling from Discrete Energy-Based Models with Quality/Efficiency Trade-offs." *arXiv preprint arXiv:2112.05702* (2021).

Algorithm 1 QRS

Require: P, q, β $\triangleright\ 0 < \beta < \infty$
while True **do**
  $x \sim q$
  $r_x \leftarrow \min\left(1, P(x)/(\beta q(x))\right)$ $\triangleright$ Acceptance prob.
  $u \sim U_{[0,1]}$ $\triangleright\ U_{[0,1]}$: unif. dist. over $[0, 1]$
  if $u \leq r_x$ **then**
    output x
  end if
end while



👍 QRS does not enforce $\boldsymbol{\beta}$ to be an upper-bound on P(x)/q(x)

👍 QRS allows to exploit good global proposals.

👉 QRS samples from the truncated distribution shaded in blue (P_β).

👉 By tuning $\boldsymbol{\beta}$ we trade-off **sampling efficiency** for **approximation quality**.

How can we quantify these?

Assessing Approximation Quality

- How far is the distribution we are sampling from, $p_\beta(x) = Z_\beta^{-1} P_\beta(x)$ from the target distribution $p(x) \doteq Z^{-1} P(x)$?

$$Z \doteq \sum_{x \in X} P(x) \qquad Z_\beta \doteq \sum_{x \in X} P_\beta(x)$$

Assessing Approximation Quality

- How far is the distribution we are sampling from, $p_\beta(x) = Z_\beta^{-1} P_\beta(x)$ from the target distribution $p(x) \doteq Z^{-1} P(x)$?

$$Z \doteq \sum_{x \in X} P(x) \qquad Z_\beta \doteq \sum_{x \in X} P_\beta(x)$$

- f-divergence between two (normalized) distributions :

$$D_f(p_1, p_2) \doteq \mathbb{E}_{x \sim p_2} f\left(\frac{p_1(x)}{p_2(x)}\right)$$

where *f*: $\mathbb{R}^+ \rightarrow \mathbb{R}$ is a convex function s.t. f(1) = 0.

e.g. if $f(t) = t \log t$ then we obtain $KL(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}$

if $f(t) = 1/2\, |1 - t|$ then we obtain $\mathrm{TVD}(p_1, p_2) \doteq 1/2 \sum_x |p_1(x) - p_2(x)|$

Assessing Approximation Quality

- How far is the distribution we are sampling from, $p_\beta(x) = Z_\beta^{-1} P_\beta(x)$ from the target distribution $p(x) \doteq Z^{-1} P(x)$?

$$Z \doteq \sum_{x \in X} P(x) \qquad Z_\beta \doteq \sum_{x \in X} P_\beta(x)$$

- f-divergence between two (normalized) distributions :

$$D_f(p_1, p_2) \doteq \mathbb{E}_{x \sim p_2} f\left(\frac{p_1(x)}{p_2(x)}\right)$$

where f: $\mathbb{R}^+ \rightarrow \mathbb{R}$ is a convex function s.t. f(1) = 0.

e.g. if $f(t) = t \log t$ then we obtain $KL(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}$

if $f(t) = 1/2\,|1 - t|$ then we obtain $\mathrm{TVD}(p_1, p_2) \doteq 1/2 \sum_x |p_1(x) - p_2(x)|$

- To compute $D_f(p, p_\beta)$, for every sample x, we need to compute $p(x)$ and $p_\beta(x)$:

$$p(x) \doteq Z^{-1} P(x) \qquad p_\beta(x) = Z_\beta^{-1} P_\beta(x)$$

known: $P(x)$

we need to compute: Z^{-1}, Z_β^{-1}, $P_\beta(x)$

Assessing Approximation Quality

Computing $P_\beta(x)$

Algorithm 1 QRS

Require: P, q, β ▷ $0 < \beta < \infty$
while True **do**
  $x \sim q$
  $r_x \leftarrow \min\left(1, P(x)/(\beta q(x))\right)$ ▷ Acceptance prob.
  $u \sim U_{[0,1]}$ ▷ $U_{[0,1]}$: unif. dist. over $[0, 1]$
  if $u \leq r_x$ **then**
    output x
  end if
end while

$$P_\beta(x) = q(x) \times \min\left(1, P(x)/(\beta q(x))\right)$$

$$= \min\left(q(x), P(x)/\beta\right)$$

⇒ QRS does not only approximately sample from the target distribution, but also allows to ***score*** *the obtained samples according to the sampling distribution*!

Assessing Approximation Quality

Computing partition functions through Importance Sampling

$$\sum_{x\in X} h(x) \;=\; \sum_{x\in X} q(x)\frac{h(x)}{q(x)} = \mathbb{E}_{x\sim q}\frac{h(x)}{q(x)}$$

$\{x_1,\ldots,x_N\}$ of i.i.d draws from q

$$\simeq N^{-1}\sum_{i\in[1,N]}\frac{h(x_i)}{q(x_i)}.$$

$$Z = \sum_{x\in X} P(x) = \mathbb{E}_{x\sim q}\frac{P(x)}{q(x)}$$
$$\simeq N^{-1}\sum_{i\in[1,N]}\frac{P(x_i)}{q(x_i)},$$

$$Z_\beta = \sum_{x\in X} P_\beta(x) = \mathbb{E}_{x\sim q}\frac{P_\beta(x)}{q(x)}$$
$$\simeq N^{-1}\sum_{i\in[1,N]}\frac{P_\beta(x_i)}{q(x_i)},$$

Assessing Approximation Quality

Computing divergences through Importance Sampling

$$\sum_{x\in X} h(x) = \sum_{x\in X} q(x)\frac{h(x)}{q(x)} = \mathbb{E}_{x\sim q}\frac{h(x)}{q(x)}$$

$\{x_1,\ldots,x_N\}$ of i.i.d draws from q

$$\simeq N^{-1}\sum_{i\in[1,N]}\frac{h(x_i)}{q(x_i)}.$$

$$\begin{aligned}
D_f(p,p_\beta) &= \mathbb{E}_{x\sim p_\beta} f\left(\frac{p(x)}{p_\beta(x)}\right)\\
&= \mathbb{E}_{x\sim q}\frac{p_\beta(x)}{q(x)} f\left(\frac{p(x)}{p_\beta(x)}\right)\\
&= \mathbb{E}_{x\sim q}\frac{P_\beta(x)}{Z_\beta\, q(x)} f\left(\frac{Z_\beta\, P(x)}{Z\, P_\beta(x)}\right)\\
&\simeq N^{-1}\sum_{i\in[1,N]}\frac{P_\beta(x_i)}{Z_\beta\, q(x_i)}\, f\left(\frac{Z_\beta\, P(x_i)}{Z\, P_\beta(x_i)}\right).
\end{aligned}$$

Assessing Efficiency

Computing acceptance rate

$$\mathrm{AR}_\beta = \mathbb{E}_{x\sim q} \min\left(1, \frac{P(x_i)}{\beta q(x_i)}\right)$$

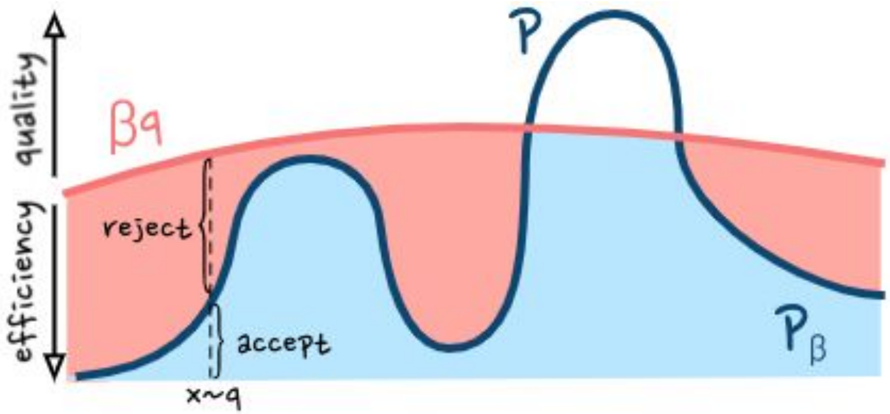
$$\simeq N^{-1} \sum_{i\in[1,N]} \min\left(1, \frac{P(x_i)}{\beta q(x_i)}\right)$$

$x \sim q$

$r_x \leftarrow \min\left(1, P(x)/(\beta q(x))\right)$ ▷ Acceptance prob.

$u \sim U_{[0,1]}$ ▷ $U_{[0,1]}$: unif. dist. over [0, 1]

Halftime Summary



👍 QRS does not enforce β to be an upper-bound on P(x)/q(x)

👉 QRS samples from the truncated distribution shaded in blue (P_β).

👉 By tuning β we trade-off **sampling efficiency** for **approximation quality**.

👍 **We can compute quality** in terms of f-divergence between $p_\beta(x)$ and p(x).

👍 **We can compute efficiency** in terms of acceptance rate.

Experiment 1: Proposal distributions

$$P(x) \doteq a(x)\mathbb{I}[\text{“wikileaks”} \in x]$$

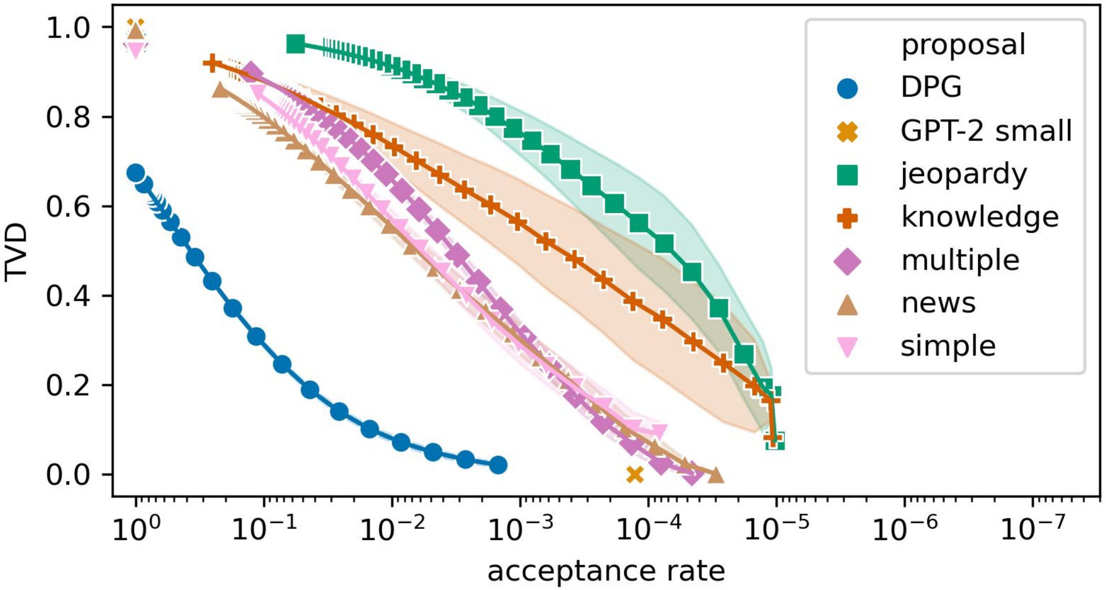
$a(x)$ → GPT-2 small

- Using alternative Proposals:
  - GPT-2 small (pure RS)
  - Prompts
  - Fine-tuned with DPG

prompt-name	prompt
simple	Wikileaks.
multiple	Wikileaks, Wikileaks, Wikileaks.
knowledge	Here is what I know about Wikileaks:
jeopardy	This medium was founded by Julian Assange in 2006.
news	Here are the latest developments on Wikileaks:

- Plot TVD (quality) as a function of acceptance rate (efficiency) for different values of β corresponding to acceptance rates in the range 1 - 10^{-5} using 1M samples from q.

Experiment 1: Proposal distributions



prompt-name	prompt
simple	Wikileaks.
multiple	Wikileaks, Wikileaks, Wikileaks.
knowledge	Here is what I know about Wikileaks:
jeopardy	This medium was founded by Julian Assange in 2006.
news	Here are the latest developments on Wikileaks:

Experiment 2: Debiasing Scientists Biographies

Scientist

$$\mathbb{E}_{x \sim p} \phi_1(x) = \bar{\mu}_1 \quad 100\%$$

$$\mathbb{E}_{x \sim p} \phi_2(x) = \bar{\mu}_2 \quad 50\%$$

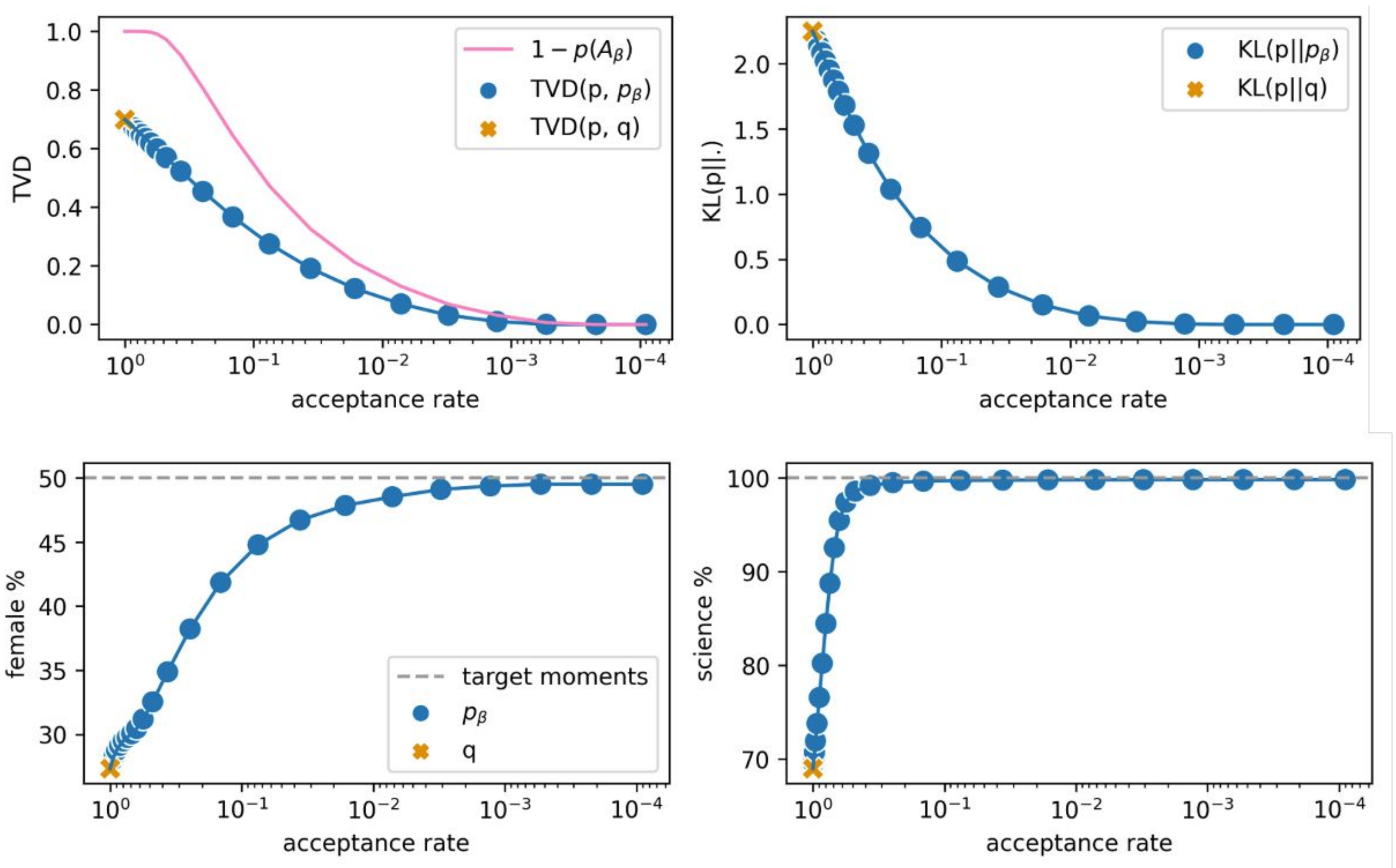
Female

$$P(x) \doteq a(x) e^{\lambda \cdot \phi(x)}$$

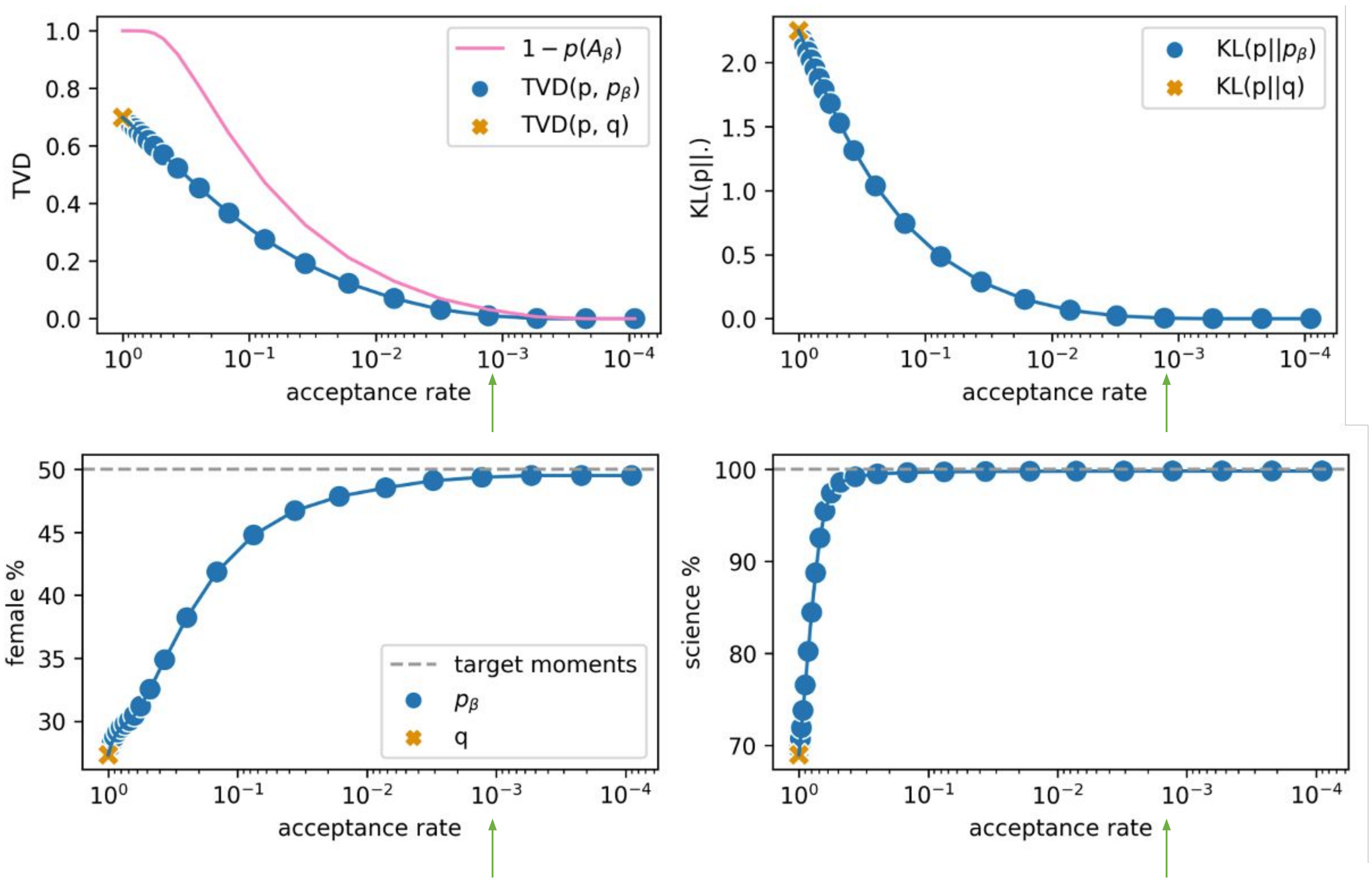
Moment Matching

- a(x) produces
  - female character biographies 7.5% of the time and
  - scientist biographies 1.8% of the time.
- As proposal distribution we use the fine-tuned (DPG) model from (Khalifa et al., 2021) that produces
  - 27.3% female biographies and
  - 69.0% scientist biographies.

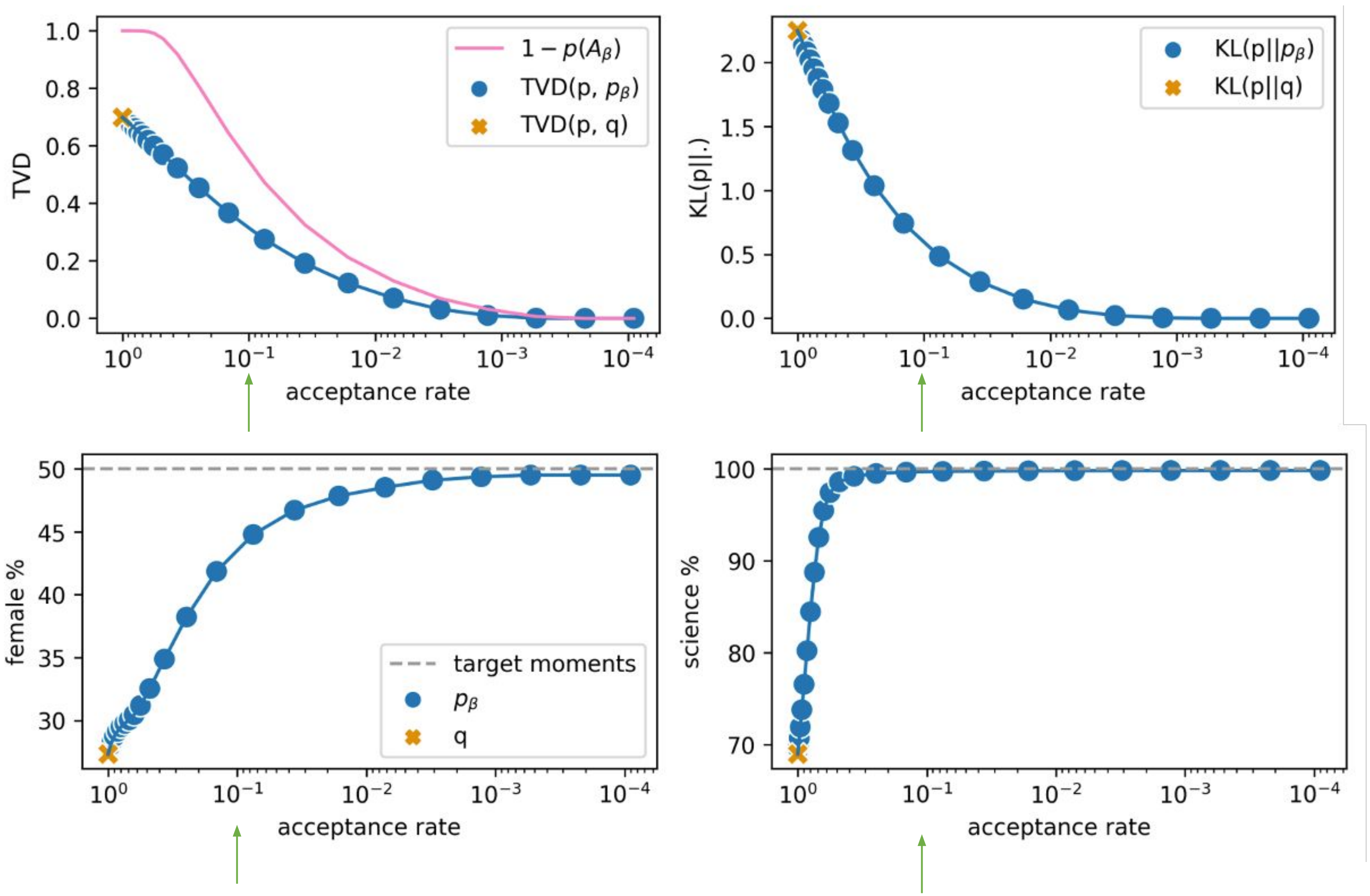
Experiment 2: Debiasing Scientists Biographies



Experiment 2: Debiasing Scientists Biographies



Experiment 2: Debiasing Scientists Biographies



Experiment 2: Debiasing Scientists Biographies

QRS samples from p at $\text{AR} = 10^{-3}$
Chandra Pradha Towni (born February 11, 1965) is a social scientist, activist, poet, and author living in Portugal. She is...
Enrella Carrière is a Canadian writer, translator, and philosopher specializing in the history of show business. She has covered topics such as the direction and psychology of television and the evolution of human...
Albert Fahn (born 1970) is an American scientist who focuses on algorithms for generating biomechanical data. Methods to generate and construct biomechanical data from...
Wyndham Radnor (born 1946) is a British historian and criminologist specialising in the subject of labour law. He has written extensively on...

Experiment 3: Comparison to MCMC

- Metropolis-Hastings (RWMH): Uses a local proposal g(x'|x) to generate a random walk across the sample space.
  - Pick an initial point x_0.
  - At every time t=0…:
    - Generate a random candidate according to $g(x' \mid x_t)$
    - Calculate acceptance probability $A(x', x_t) = \min\left(1, \frac{P(x')}{P(x_t)} \frac{g(x_t \mid x')}{g(x' \mid x_t)}\right)$
    - If accept, set $x_{t+1} = x'$. Otherwise, $x_{t+1} = x_t$.

- Independence Metropolis Hastings (IMH):

  Uses a global proposal: $g(x'|x_t) = q(x')$

Experiment 3: Comparison to MCMC

$$P(x) = a(x)\mathbb{I}[\text{“amazing”} \in x]$$

RWMH/IMH:

👉 We use the DPG fine-tuned global proposal q(x) to initialize the chain.

👉 For RWMH we use as local proposal g(x'|x) that deletes/inserts/edit a token using a mix of BERT and a dirac-delta that adds the word "amazing".

👉 For IMH, we use the DPG fine-tuned as the global proposal q(x').

👉 We use a burn-in period of 1000 samples, and then keep 1 out 1000 samples.

👉 We also have reset variants, in which we re-initialize the chain from the global proposal.

QRS:

👉 We use the DPG fine-tuned global proposal q(x)

👉 We set an acceptance rate of 10^{-3}.

Experiment 3: Comparison to MCMC

- Evaluation:
  - MCMC does not provide probability scores for the samples it produces. As such, ⚠️ there is no obvious way to compute divergences with MCMC!
  - Convergence diagnostics for MCMC is a difficult and debated topic[1].
  - We resort to "proxy" metrics (*but these can be cheated*!)
    - Constraint satisfaction
    - Perplexity
    - Self-BLEU-5 } Diversity across samples
    - %Uniq } Diversity across samples
    - Dist-2 } Diversity within samples

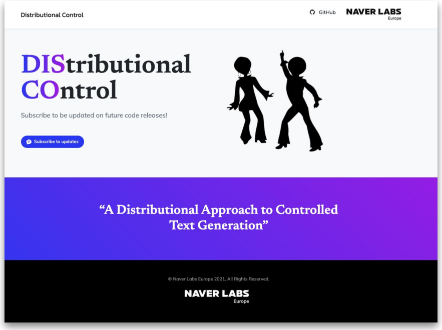
[1] Vivekananda Roy. *Convergence diagnostics for markov chain monte carlo.* Annual Review of Statistics and Its Application, 7(1):387-412, 2020.

Experiment 3: Comparison to MCMC

Method	%Amazing	PPL↓	Self-BLEU-5↓	%Uniq↑	Dist-2↑	TVD↓*
proposal	62.9 ± 0.4	61.7 ± 0.3	85.8 ± 0.1	100 ± 0.0	96.1 ± 0.0	0.67
RWMH	100 ± 0.0	-	99.8 ± 0.2	32.0 ± 33.7	83.8 ± 17	Unk.
RWMH-R	100 ± 0.1	58.7 ± 3.3	87.6 ± 0.4	100 ± 0.0	92.0 ± 0.3	Unk.
IMH	100 ± 0.0	-	86.9 ± 0.3	98.7 ± 0.5	96.3 ± 0.1	Unk.
IMH-R	100 ± 0.0	63.4 ± 1.5	86.7 ± 0.1	100 ± 0.0	96.3 ± 0.1	Unk.
QRS	100 ± 0.0	62.8 ± 1.6	86.6 ± 0.2	100 ± 0.0	96.3 ± 0.1	0.01

In summary

https://disco.naverlabs.com/



- From many practical problems, we need to generate samples from an EBM.
- Provided a global proposal, **QRS** can **approximate the target distribution to any desired level of quality in exchange for sampling efficiency.**
- Crucially, **we can quantity the trade-off between quality (in terms of f-divergences)** and efficiency (in terms of acceptance rate) to decide the level that is most appropriate for our goals.
- High-quality global proposals are becoming increasingly more available thanks to advancements in deep learning (e.g., DPG, prompting, back-translation).
- QRS also performs better than local variants RWMH) and on-par to IMH (when resetting the chain) assessed in terms of *proxy metrics*, but QRS provides *divergence estimates*, whereas IMH typically does not.

Q/A