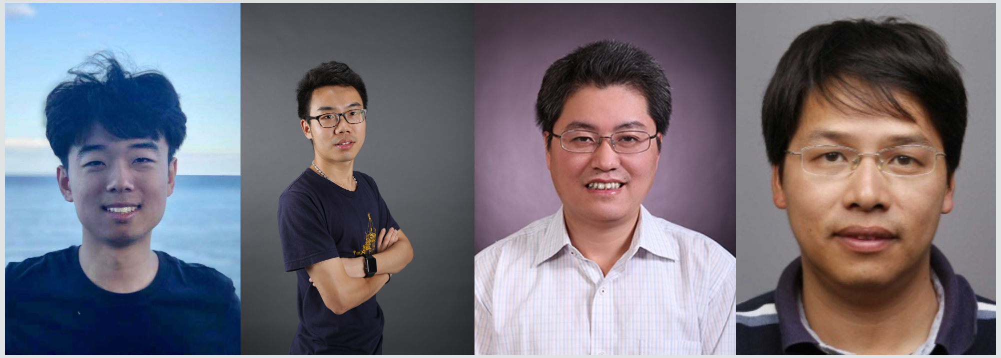
ECML PKDD 2021

北京大學
PEKING UNIVERSITY

ZERO LAB

Reparameterized Sampling for Generative Adversarial Networks

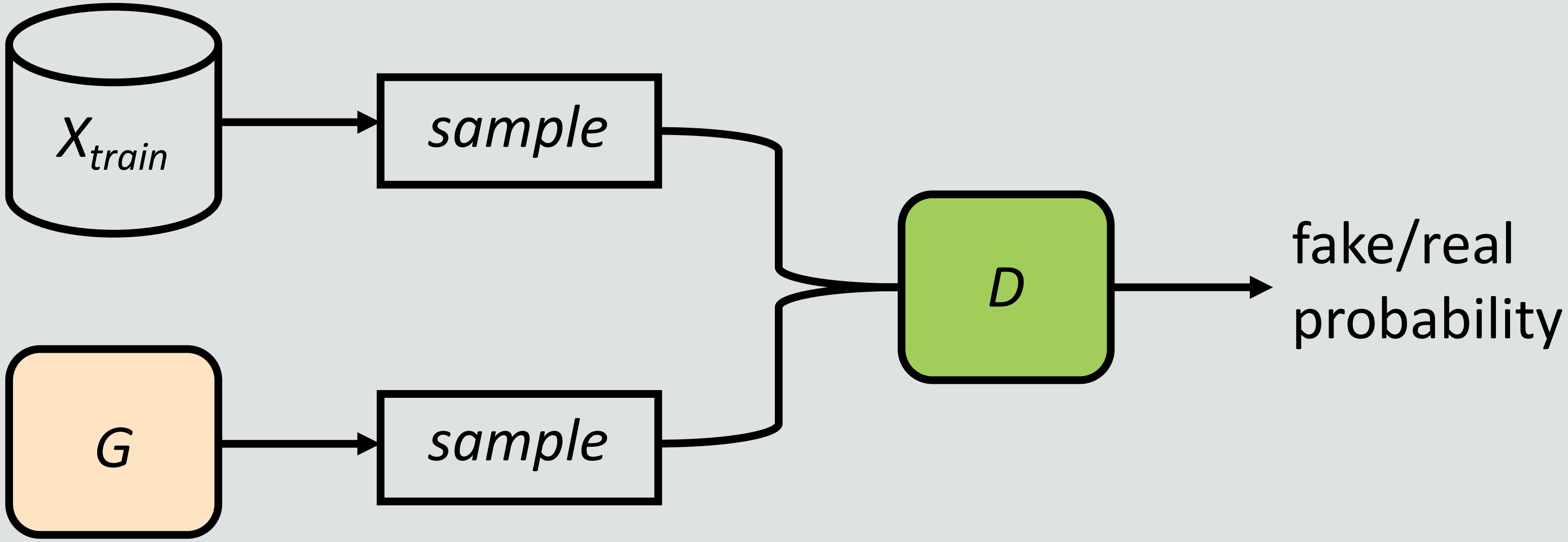
Yifei Wang, Yisen Wang, Jiansheng Yang, Zhouchen Lin



https://yifeiwang77.github.io

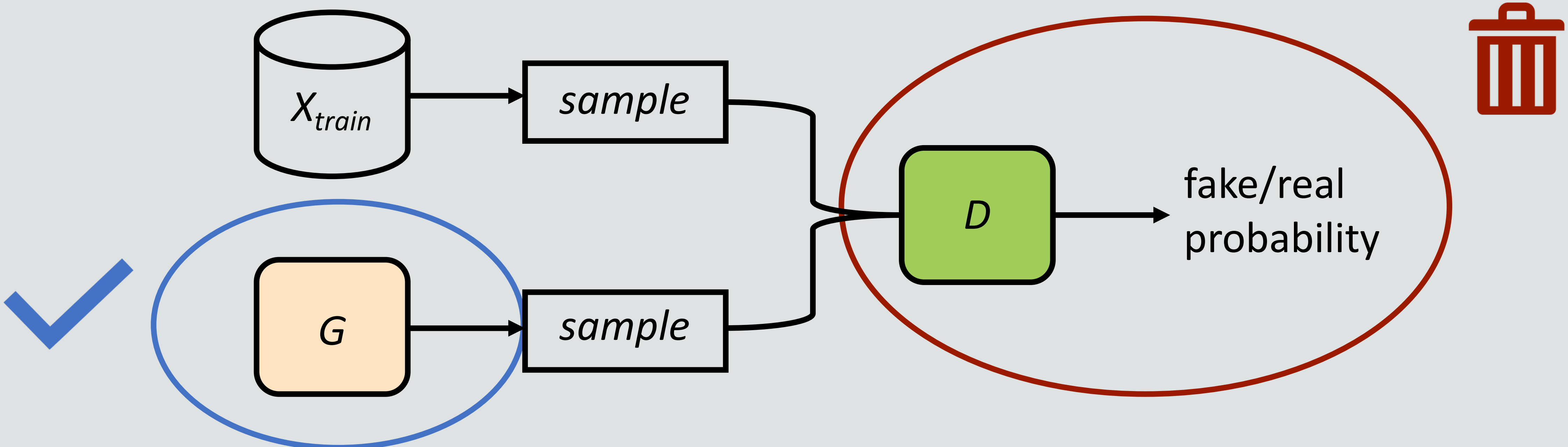
Background

- GANs learn to generate images with an adversarial game
  - between a generator (G) and a discriminator (D)



Background

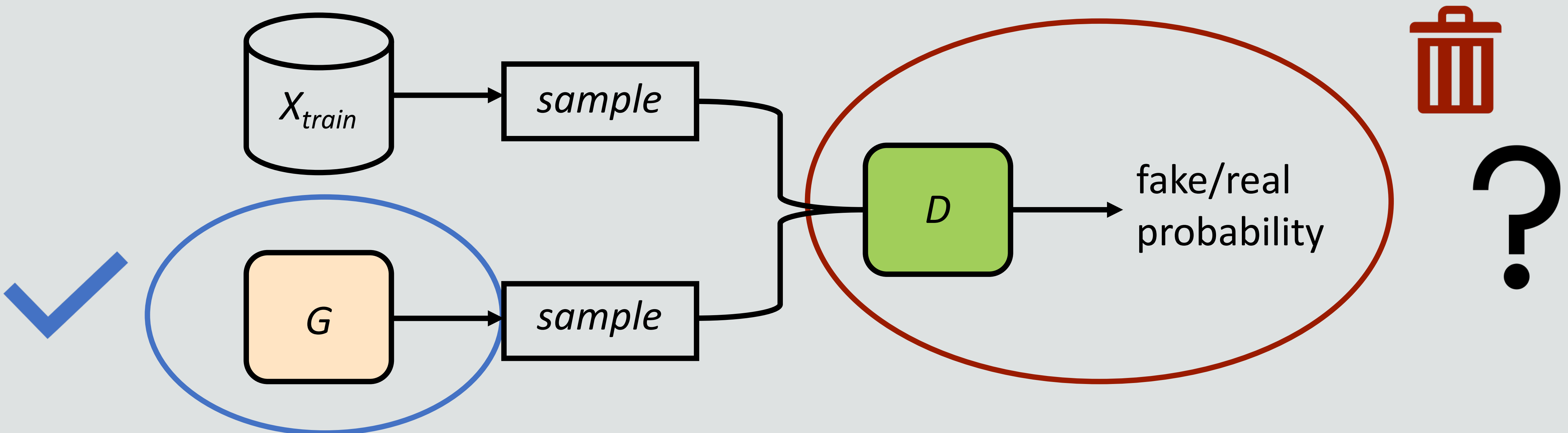
- GANs learn to generate images with an adversarial game
  - between a generator (G) and a discriminator (D)
- After training, the discriminator is thrown away, and only the generator is left for generating images



Background

- **But wait!**

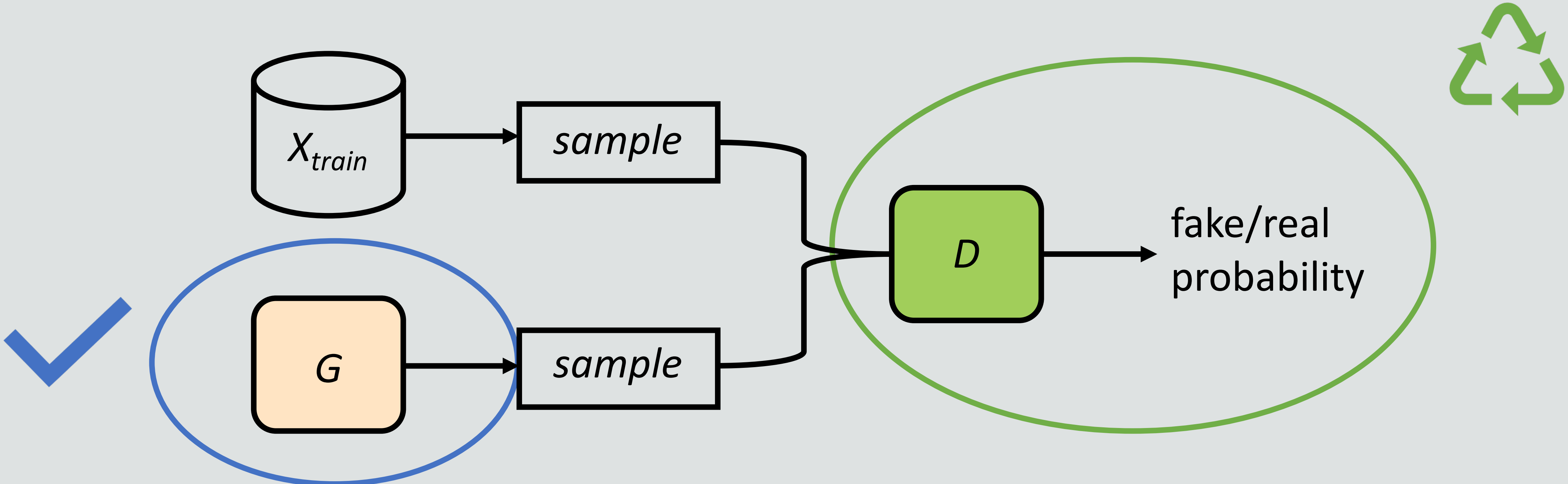
Is the discriminator (D) really useless?



Background

- **But wait!**

We can use D to further improve sample quality!



Wasted Wealth in the Discriminator

- Goal: approximating data distribution $p_d(x)$
- What we have: (imperfect) generator distribution $p_g(x)$
- Goodfellow et al. (2014): a perfect D learns density ratio

$$D(\mathbf{x}) = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \quad \Rightarrow \quad \frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} = \frac{1}{D(\mathbf{x})^{-1} - 1}.$$

- Leveraging this information in D, we can further bridge the gap between $p_g(x)$ and $p_d(x)$ and get closer to the data distribution!

Bridging the distribution gap with MCMC

- A natural solution is MCMC (Markov chain Monte Carlo)
  - starts from the initial distribution $p_0(x)=p_g(x)$
  - gradually converges to the target distribution $p_t(x)= p_d(x)$
- Metropolis-Hastings (MH) algorithm
  - 1. initial state x_0: draw a sample from the generator $p_g(x)$
  - 2. draw a proposal x' from a proposal distribution $q(x'|x_k)$
  - 3. MH-test: accept x' by flipping a coin with probability $\alpha(x', x_k)$, which is knowns as the MH acceptance ratio, or MH ratio

$$\alpha(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_t(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_t(\mathbf{x}_k)q(\mathbf{x}'||\mathbf{x}_k)}\right) \in [0, 1].$$

  - if x' is accepted, we have $x_{k+1}=x'$
  - if x' is rejected, we have $x_{k+1}=x_k$

Bridging the distribution gap with MCMC

- A natural solution is MCMC (Markov chain Monte Carlo)
  - starts from the initial distribution $p_0(x)=p_g(x)$
  - gradually converges to the target distribution $p_t(x)= p_d(x)$
- Metropolis-Hastin[illegible]lgorithm

Problem 1

  - 1. initial st[illegible]a sample from the generator $p_g(x)$
  - 2. draw a proposal x' from ***a proposal distribution $q(x'|x_k)$***
  - 3. MH-test: accept x' by flipping a coin with probability $\alpha(x', x_k)$, which is knowns as the MH acceptance ratio, or MH ratio

$$\alpha(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_t(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_t(\mathbf{x}_k)q(\mathbf{x}'||\mathbf{x}_k)}\right) \in [0, 1].$$

    - if x' is accepted, we have $x_{k+1}=x'$
    - if x' is rejected, we have $x_{k+1}=x_k$

Bridging the distribution gap with MCMC

- A natural solution is MCMC (Markov chain Monte Carlo)
  - starts from the initial distribution $p_0(x)=p_g(x)$
  - gradually converges to th… target distribution $p_t(x)= p_d(x)$
- Metropolis-Hasti… …lgorithm
  - 1. …al st… …a sample from the generator $p_g(x)$
  - … a pr…osal x' from ***a proposal distribution $q(x'|x_k)$***
  - 3. MH-test: accept x' by ***flipping a coin with probability $\alpha(x', x_k)$,*** which is knowns as the MH acceptance ratio, or MH ratio

$$\alpha(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_t(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_t(\mathbf{x}_k)q(\mathbf{x}'||\mathbf{x}_k)}\right) \in [0, 1].$$

    - if x' is accepted, we have $x_{k+1}=x'$
    - if x' is rejected, we have $x_{k+1}=x_k$

Problem 1

Problem 2

MH-GAN and its Limitations

- Problem 1: MH-GAN adopts an independent proposal, i.e.,

$$\mathbf{x}' \sim q\left(\mathbf{x}' | \mathbf{x}_k\right) = q\left(\mathbf{x}'\right) = p_g(\mathbf{x}').$$

- Problem 2: it admits a tractable MH ratio,

$$\alpha_{\mathrm{MH}}\left(\mathbf{x}', \mathbf{x}_k\right) = \min\left(1, \frac{p_d(\mathbf{x}')q(\mathbf{x}_k)}{p_d(\mathbf{x}_k)q(\mathbf{x}')}\right) = \min\left(1, \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1}\right).$$

- ***Achilles‘ heel**: sample inefficiency due to independent proposal*
  - acceptance ratio could be very low (<5% in practice)
  - the chain can be trapped for a very long time

Turner, R., Hung, J., Frank, E., Saatchi, Y. and Yosinski, J. Metropolis-Hastings Generative Adversarial Networks. In *ICML.. 2019.*

Improving Sample Efficiency...But How?

- It is natural to consider a ***dependent*** (DEP) proposal $q(x'|x_k)$
- Two problems occur:
- 1) Hard to design proposals in the ***high-dimensional space*** $\mathcal{X}$
  - complex, highly non-convex landscape is hard to explore
- 2) The MH ratio is ***no longer tractable***!

$$\alpha_{\text{DEP}}(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_d(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_d(\mathbf{x}_k)q(\mathbf{x}'|\mathbf{x}_k)}\right),$$

  - $p_d(x)$ is unknown!

Improving Sample Efficiency...But How?

- It is natural to consider a ***dependent*** (DEP) proposal $q(x'|x_k)$
- Two problems occur:
- 1) Hard to design proposals in the ***high-dimensional space $\mathcal{X}$***
  - complex, highly non-convex landscape is hard to explore
- 2) The MH ratio is ***no longer tractable***!

$$\alpha_{\mathrm{DEP}}(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_d(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_d(\mathbf{x}_k)q(\mathbf{x}'|\mathbf{x}_k)}\right),$$

  - $p_d(x)$ is unknown!

Does it puts dependent proposals to death? NO!

Our Solution: Transition in the Latent Space!

- In GANs, we learn to map from a ***low-dimensional latent space*** $\mathcal{Z}$ to a ***high-dimensional sample space*** $\mathcal{X}$ with the generator *G*

$$\mathbf{x} = G(\mathbf{z}), \quad \mathbf{z} \sim p_0(\mathbf{z}),$$

  - leveraging structural information to design better sampling trajectories
- Insight: it will be a lot ***easier*** to design **transitions in the latent space**
  - a structured proposal with lower dimensionality & simpler geometry
- Perhaps surprisingly, it also leads to a ***tractable MH ratio!***

Our Solution: Transition in the Latent Space!

- In GANs, we learn to map from a ***low-dimensional latent space*** $\mathcal{Z}$ to a ***high-dimensional sample space*** $\mathcal{X}$ with the generator *G*

$$\mathbf{x} = G(\mathbf{z}), \quad \mathbf{z} \sim p_0(\mathbf{z}),$$

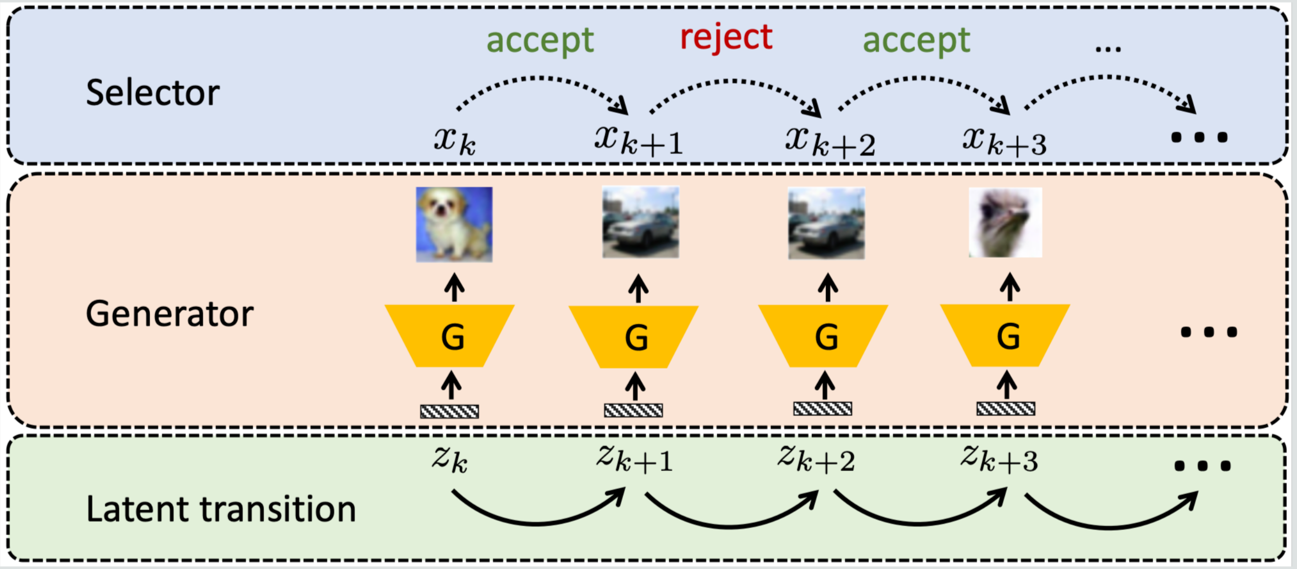
  - leveraging structural information to design better sampling trajectories
- Insight: it will be a lot ***easier*** to design **transitions in the latent space**
  - a structured proposal with lower dimensionality & simpler geometry
- Perhaps surprisingly, it also leads to a ***tractable MH ratio!***

Method: reparameterizating q(x’|x) -> q(z’|z)

$$\log q_{\mathrm{REP}}(\mathbf{x}'|\mathbf{x}_k) = \log q(\mathbf{x}'|\mathbf{z}_k) = \log q(\mathbf{z}'|\mathbf{z}_k) - \frac{1}{2}\log\det J_{\mathbf{z}'}^\top J_{\mathbf{z}'},$$

REParameterized (REP) Proposal

- It reparameterizes $q_{REP}(x'|x_k)$ with two coupling Markov chains
  - latent-space Markov chain: ***draw a latent proposal*** z' from $q(z'|z_k)$
  - generator: ***push the latent z' forward*** and get sample proposal $x'=G(z')$
  - sample-space Markov chain: ***decide the acceptance*** of $x'=G(z')$



Tractable MH criterion

- The following theorem shows that our REP proposal admits a ***tractable MH ratio for general latent proposals $q(z'|z_k)$***

Theorem 1. *Consider a Markov chain of GAN samples $\mathbf{x}_{1:K}$ with initial distribution $p_g(\mathbf{x})$. For step $k+1$, we accept our REP proposal $\mathbf{x}' \sim q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k)$ with probability*

$$\alpha_{\text{REP}}(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{p_0(\mathbf{z}')q(\mathbf{z}_k|\mathbf{z}')}{p_0(\mathbf{z}_k)q(\mathbf{z}'|\mathbf{z}_k)} \cdot \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1}\right), \tag{9}$$

i.e. let $\mathbf{x}_{k+1} = \mathbf{x}'$ if $\mathbf{x}'$ is accepted and $\mathbf{x}_{k+1} = \mathbf{x}_k$ otherwise. Further assume the chain is irreducible, aperiodic and not transient. Then, according to the Metropolis-Hastings algorithm, the stationary distribution of this Markov chain is the data distribution $p_d(\mathbf{x})$ [6].

  - it also reduces to MH-GAN's MH ratio (as a special case) when adopting an independent proposal $q(z'|z)=q(z')$

Proof Sketch

- Change of variables due to reparameterization
  - the generator $\log p_g(\mathbf{x})|_{\mathbf{x}=G(\mathbf{z})} = \log p_0(\mathbf{z}) - \frac{1}{2}\log\det J_{\mathbf{z}}^{\top} J_{\mathbf{z}}.$
  - the proposal $\log q_{\mathrm{REP}}(\mathbf{x}'|\mathbf{x}_k) = \log q(\mathbf{x}'|\mathbf{z}_k) = \log q(\mathbf{z}'|\mathbf{z}_k) - \frac{1}{2}\log\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'},$
- Combined into the MH acceptance

$$
\begin{aligned}
\alpha_{\mathrm{REP}}(\mathbf{x}', \mathbf{x}_k) = \frac{p_d(\mathbf{x}')\, q(\mathbf{x}_k|\mathbf{x}')}{p_d(\mathbf{x}_k)\, q(\mathbf{x}'|\mathbf{x}_k)} &= \frac{p_d(\mathbf{x}') q(\mathbf{z}_k|\mathbf{z}') \left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}} p_g(\mathbf{x}_k) p_g(\mathbf{x}')}{p_d(\mathbf{x}_k) q(\mathbf{z}'|\mathbf{z}_k) \left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}} p_g(\mathbf{x}') p_g(\mathbf{x}_k)} \\
&= \frac{q(\mathbf{z}_k|\mathbf{z}')\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}} p_0(\mathbf{z}')\left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}} (D(\mathbf{x}_k)^{-1} - 1)}{q(\mathbf{z}'|\mathbf{z}_k)\left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}} p_0(\mathbf{z}_k)\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}} (D(\mathbf{x}')^{-1} - 1)} \\
&= \frac{p_0(\mathbf{z}') q(\mathbf{z}_k|\mathbf{z}') (D(\mathbf{x}_k)^{-1} - 1)}{p_0(\mathbf{z}_k) q(\mathbf{z}'|\mathbf{z}_k) (D(\mathbf{x}')^{-1} - 1)},
\end{aligned}
$$

Case Study: Latent Langevin Monte Carlo

- We can use gradients to explore the landscape more efficiently
- Sample-level Langevin Monte Carlo (LMC)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\tau}{2}\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_k) + \sqrt{\tau} \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

  - is intractable because $p_t(x)=p_d(x)$ is unknown
- Latent Langevin Monte Carlo (L2MC) is tractable w/ reparameterization!

$$\begin{aligned}
\mathbf{z}' &= \mathbf{z}_k + \frac{\tau}{2}\nabla_{\mathbf{z}} \log p_t(\mathbf{z}_k) + \sqrt{\tau} \cdot \varepsilon \\
&= \mathbf{z}_k + \frac{\tau}{2}\nabla_{\mathbf{z}} \log \frac{p_t(\mathbf{z}_k)\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}}}{p_0(\mathbf{z}_k)\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}}} + \frac{\tau}{2}\nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \varepsilon \\
&= \mathbf{z}_k + \frac{\tau}{2}\nabla_{\mathbf{z}} \log \frac{p_d(\mathbf{x}_k)}{p_g(\mathbf{x}_k)} + \frac{\tau}{2}\nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \varepsilon \\
&= \mathbf{z}_k - \frac{\tau}{2}\nabla_{\mathbf{z}} \log(D^{-1}(\mathbf{x}_k) - 1) + \frac{\tau}{2}\nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),
\end{aligned}$$

A Unified Framework for GAN Sampling

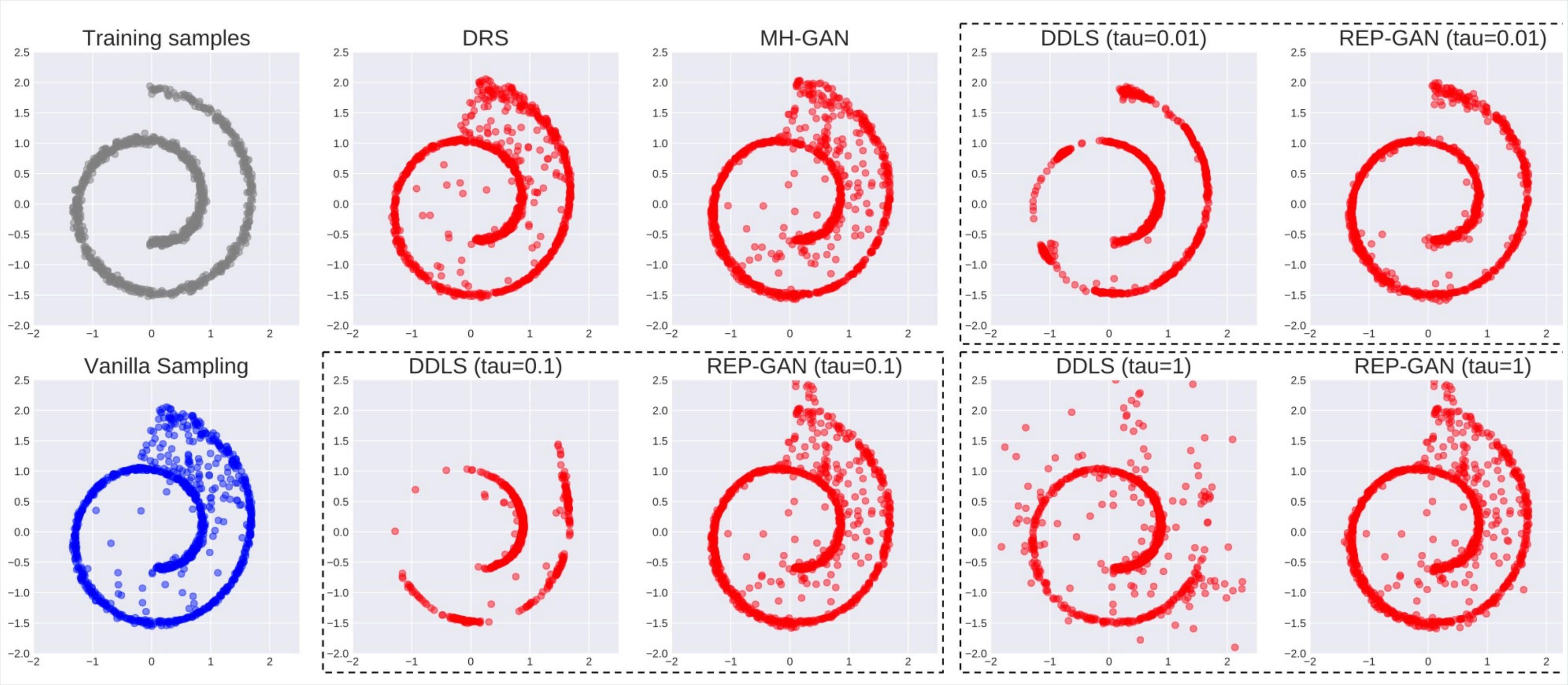
- REP-GAN: an efficient sampling method for GANs (also work for WGAN)
  - REP proposal that works for general latent dependent proposals
  - Tractable MH ratio $\alpha_{REP}(x', x_k)$
  - A practical latent proposal: L2MC
- It serves as a general recipe for GAN sampling, as we take previous work as our special cases

Table 1: Comparison of sampling methods for GANs in terms of three effective sampling mechanisms.

Method	Rejection step	Markov chain	Latent gradient proposal
GAN	✗	✗	✗
DRS [2]	✓	✗	✗
MH-GAN [27]	✓	✓	✗
DDLS [5]	✗	✓	✓
REP-GAN (ours)	✓	✓	✓

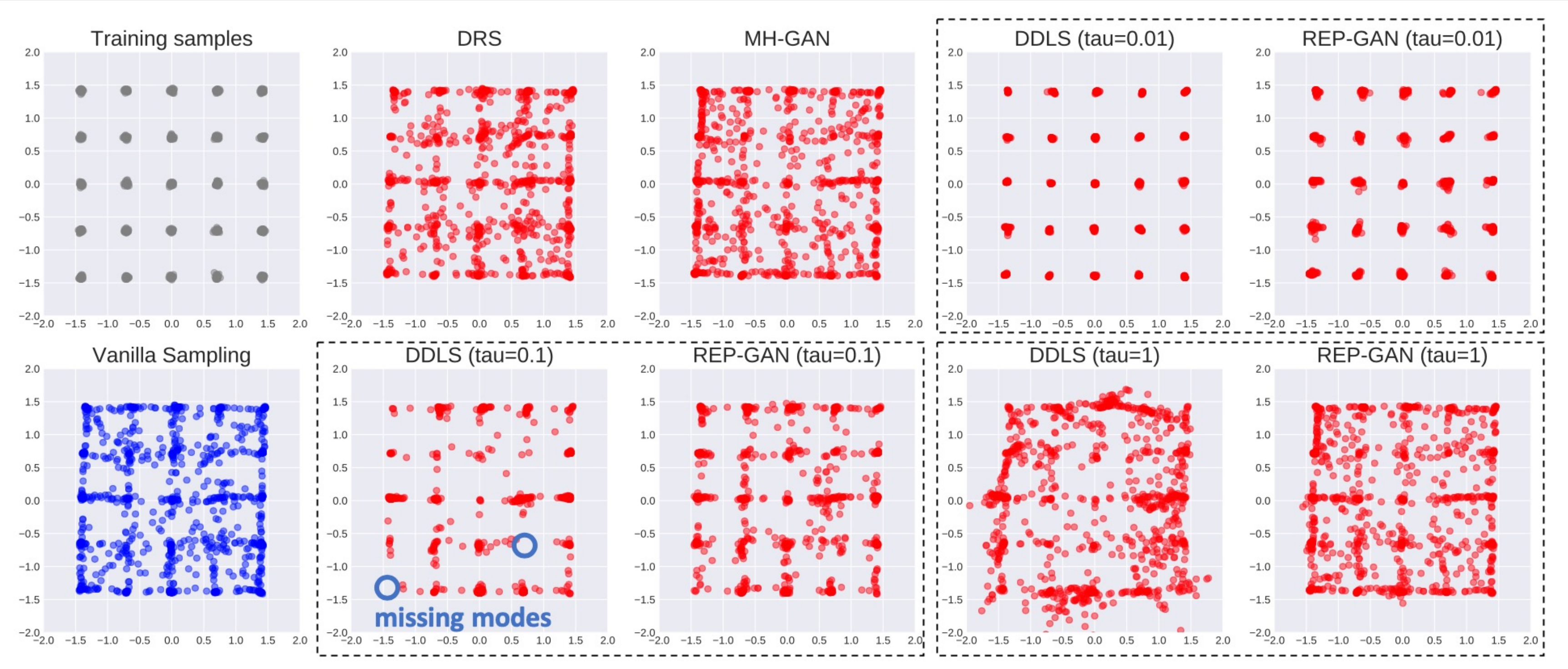
Experiments on Synthetic Datasets

- Manifold learning of Swiss Roll
  - Less discontinuous points
  - More robust to step size



Experiments on Synthetic Datasets

- Multi-modal Experiments of Mixture of Gaussians
  - Less missing modes
  - More robust to step size



Experiments on Real-world Datasets

- CIFAR-10 and CelebA with DCGAN and WGAN
  - **Clear improvement of sample quality**

Table 2: Inception Scores of different sampling methods on CIFAR-10 and CelebA, with the DCGAN and WGAN backbones.

Method	CIFAR-10		CelebA	
	DCGAN	WGAN	DCGAN	WGAN
GAN	3.219	3.740	2.332	2.788
DRS [2]	3.073	3.137	2.869	2.861
MH-GAN [27]	3.225	3.851	**3.106**	2.889
DDLS [5]	3.152	3.547	2.534	2.862
REP-GAN (ours)	**3.541**	**4.035**	2.686	**2.943**

Experiments on Real-world Datasets

- CIFAR-10 and CelebA with DCGAN and WGAN
  - Clear improvement of sample quality
  - **Significantly improved sample efficiency**
    - average acceptance ratio: 5% -> around 40%

Table 3: Average Inception Score (a) and acceptance ratio (b) vs. training epochs with DCGAN on CIFAR-10.

(a) Inception Score (mean $\pm$ std)

Epoch	20	21	22	23	24
GAN	2.482 ± 0.027	3.836 ± 0.046	3.154 ± 0.014	3.383 ± 0.046	3.219 ± 0.036
MH-GAN	2.356 ± 0.023	3.891 ± 0.040	3.278 ± 0.033	3.458 ± 0.029	3.225 ± 0.029
DDLS	2.419 ± 0.021	3.332 ± 0.025	2.996 ± 0.035	3.255 ± 0.045	3.152 ± 0.028
REP-GAN	$\mathbf{2.487} \pm 0.019$	$\mathbf{3.954} \pm 0.046$	$\mathbf{3.294} \pm 0.030$	$\mathbf{3.534} \pm 0.035$	$\mathbf{3.541} \pm 0.038$

(b) Average Acceptance Ratio (mean $\pm$ std)

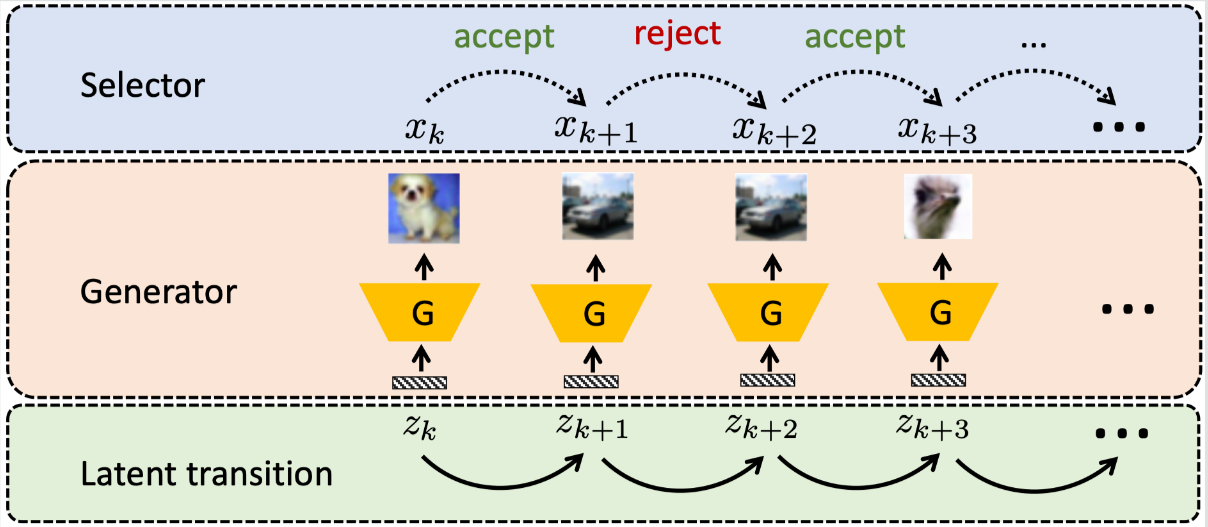
Epoch	20	21	22	23	24
MH-GAN	0.028 ± 0.143	0.053 ± 0.188	0.060 ± 0.199	0.021 ± 0.126	0.027 ± 0.141
REP-GAN	$\mathbf{0.435} \pm 0.384$	$\mathbf{0.350} \pm 0.380$	$\mathbf{0.287} \pm 0.365$	$\mathbf{0.208} \pm 0.335$	$\mathbf{0.471} \pm 0.384$

Takeaways

- **GANs:** both D and G contain useful information to cultivate
- **Variational inference:** sampling methods can be used to further bridge the variational distribution and the data distribution
- **Sampling:** low-dimensional latent space is easier to play around, and enjoys better sample efficiency
- **MCMC:** transition reparameterization for implicit models (like GANs) can also be tractable

Thanks!

Q & A



For more details, please refer to our paper: https://arxiv.org/abs/2107.00352
More interesting papers @ PKU ZERO lab: https://zero-lab-pku.github.io/
Contact:
yifei_wang AT pku.edu.cn; yisen.wang AT pku.edu.cn