Guiding Deep Molecular Optimization With Genetic Exploration

Sungsoo Ahn; Junsu Kim; Hankook Le; Junwoo Shin

Korea Advanced Institute of Science & Technology

Gunnar Beck Nelson-Virginia Tech

Disclaimer

404 error github

Overview

- Mission
- Introduction
- Related Work
- Solution (P1, P2, P3)
- Experiments (P1, P2, P3)
- Conclusion & Broader Impacts
- Discussion

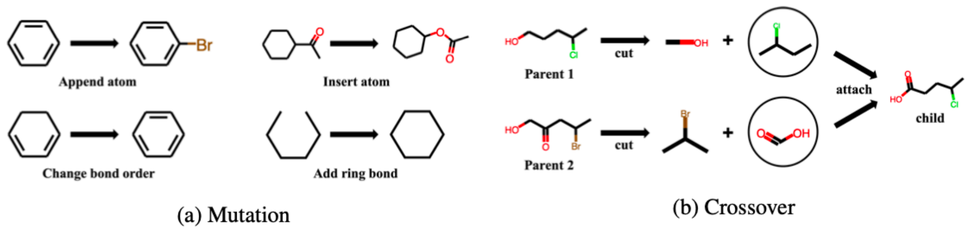
Mission:

Focus on improving “de novo” molecular design

The ability to create new molecular structures from scratch for useful application (Drug discovery, material design, etc)

Disclaimer

Molecules are unpredictable in nature due to mutations and crossovers



(a) Mutation (b) Crossover

Figure 2: Illustration of (a) mutation and (b) crossover used in the genetic expert policy.

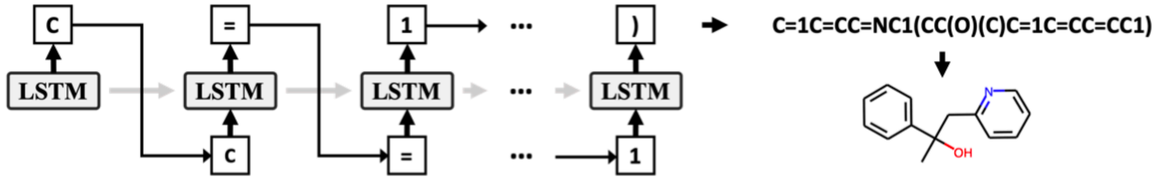


Figure 3: Illustration of the apprentice policy generating a SMILES representation of a molecule.

Question

How can we control nature?

Introduction

- Deep Neural Networks (DNNs)
  - demonstrated successful results for solving de novo molecular design
- Other related works focus on the problem of drug design creation
- The question:
  - As there are new forefronts with DNNs with optimizing the outcomes of the desired molecular structure
  - How can we validate the molecular structure for its desired outcome?

Related Work

Other technology explored

- Deep Reinforcement Learning:
  - Model Adaptation = Reward of desired outcomes
  - Applications: Protein Structure Design, Biological Sequences
- Deep Embedding Optimization:
  - Having neural networks learn from characteristics (Embeddings) from DNNs and apply them on a universal stand
  - Applications: Particle Swarm Optimization, Bayesian Optimizations
- Genetic Algorithms:
  - Search over molecular space, looking over mutations and crossover (Evolution based)
  - Applications: Optimization problems

Solution

Genetic Expert Guided Learning (GEGL)

- Deep Reinforcement Learning Framework
- Premise:
  - To award the Desired outcome of the molecular design
  - From the initial (Apprentice) stage to the final (expert) stage
- 3 steps:
  - 1.) Apprentice stage: the DNN takes in the molecules with a given space and compares
  - 2.) Expert Policy: Checks the queue to match what the desired molecule is in a given space
  - 3.) The parameters are optimized in the apprentice stage to obtained the desired outcome

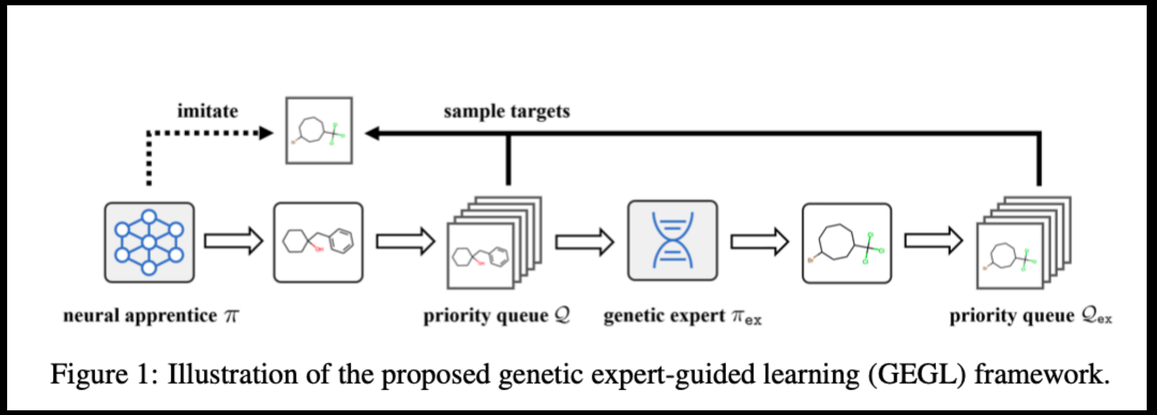


Figure 1: Illustration of the proposed genetic expert-guided learning (GEGL) framework.

Solution (P2)

Genetic Expert Guided Learning (GEGL)

Deep Reinforcement Learning Framework

- x a given molecule
- $r(x)$ The desired reward molecule
- Θ The DNN's Parameters (able to adjust within a space
- Q Queue
- Q_{ex} The Expert Queue
- $Q \cup Q_{ex}$ The Union of Apprentice Queue and Expert Queue
- $\pi(x;\Theta)$ The neural apprentice policy
- $\pi_{ex}(x;Q)$ The expert apprentice policy
- K Constant representing size
- $\sum_{x \in Q \cup Q_{ex}} log\pi(x;\Theta)$ The sum of all policies the molecule endured for the desired max output

Solution (P3)

Genetic Expert Guided Learning (GEGL)

- Deep Learning Framework:
- Set at initial stage (with each instance of time)
- As there are more samples
- Queue is being updated with each policy, checking for desired output
- While new samples are generated, the queue will be updated for each molecule
- While the expert queue is being updated for the size, being less tan the minimal reward function
- The algorithm ends when all policies have been met with the desired matches from the apprentice queue

Algorithm 1 Genetic expert-guided learning (GEGL)

Set $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{Q}_{\mathtt{ex}} \leftarrow \emptyset$. ▷ *Initialize the max-reward priority queues $\mathcal{Q}$ and $\mathcal{Q}_{\mathtt{ex}}$.*
for $t = 1, \ldots, T$ **do**
  for $m = 1, \ldots, M$ **do** ▷ *Step A: add M samples generated by π into $\mathcal{Q}$.*
    Update $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\boldsymbol{x}\}$, where $\boldsymbol{x} \sim \pi(\boldsymbol{x}; \theta)$.
    If $|\mathcal{Q}| > K$, update $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{\boldsymbol{x}_{\mathtt{min}}\}$, where $\boldsymbol{x}_{\mathtt{min}} = \arg\min_{\boldsymbol{x} \in \mathcal{Q}} r(\boldsymbol{x})$.
  end for
  for $m = 1, \ldots, M$ **do** ▷ *Step B: add M samples generated by $\pi_{\mathtt{ex}}$ into $\mathcal{Q}_{\mathtt{ex}}$.*
    Update $\mathcal{Q}_{\mathtt{ex}} \leftarrow \mathcal{Q}_{\mathtt{ex}} \cup \{\boldsymbol{x}\}$, where $\boldsymbol{x} \sim \pi_{\mathtt{ex}}(\boldsymbol{x}; \mathcal{Q})$.
    If $|\mathcal{Q}_{\mathtt{ex}}| > K$, update $\mathcal{Q}_{\mathtt{ex}} \leftarrow \mathcal{Q}_{\mathtt{ex}} \setminus \{\boldsymbol{x}_{\mathtt{min}}\}$, where $\boldsymbol{x}_{\mathtt{min}} = \arg\min_{\boldsymbol{x} \in \mathcal{Q}_{\mathtt{ex}}} r(\boldsymbol{x})$.
  end for
  Maximize $\sum_{\boldsymbol{x} \in \mathcal{Q} \cup \mathcal{Q}_{\mathtt{ex}}} \log \pi(\boldsymbol{x}; \theta)$ over θ. ▷ *Step C: train π with imitation learning.*
end for
Report $\mathcal{Q} \cup \mathcal{Q}_{\mathtt{ex}}$ as the output. ▷ *Output the highly-rewarding molecules.*

Algorithm 1 Genetic expert-guided learning (GEGL)

Set $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{Q}_{\mathtt{ex}} \leftarrow \emptyset$. ▷ *Initialize the max-reward priority queues $\mathcal{Q}$ and $\mathcal{Q}_{\mathtt{ex}}$.*
for $t = 1, \ldots, T$ **do**
  for $m = 1, \ldots, M$ **do** ▷ *Step A: add M samples generated by π into $\mathcal{Q}$.*
    Update $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\boldsymbol{x}\}$, where $\boldsymbol{x} \sim \pi(\boldsymbol{x}; \theta)$.
    If $|\mathcal{Q}| > K$, update $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{\boldsymbol{x}_{\mathtt{min}}\}$, where $\boldsymbol{x}_{\mathtt{min}} = \arg\min_{\boldsymbol{x} \in \mathcal{Q}} r(\boldsymbol{x})$.
  end for
  for $m = 1, \ldots, M$ **do** ▷ *Step B: add M samples generated by $\pi_{\mathtt{ex}}$ into $\mathcal{Q}_{\mathtt{ex}}$.*
    Update $\mathcal{Q}_{\mathtt{ex}} \leftarrow \mathcal{Q}_{\mathtt{ex}} \cup \{\boldsymbol{x}\}$, where $\boldsymbol{x} \sim \pi_{\mathtt{ex}}(\boldsymbol{x}; \mathcal{Q})$.
    If $|\mathcal{Q}_{\mathtt{ex}}| > K$, update $\mathcal{Q}_{\mathtt{ex}} \leftarrow \mathcal{Q}_{\mathtt{ex}} \setminus \{\boldsymbol{x}_{\mathtt{min}}\}$, where $\boldsymbol{x}_{\mathtt{min}} = \arg\min_{\boldsymbol{x} \in \mathcal{Q}_{\mathtt{ex}}} r(\boldsymbol{x})$.
  end for
  Maximize $\sum_{\boldsymbol{x} \in \mathcal{Q} \cup \mathcal{Q}_{\mathtt{ex}}} \log \pi(\boldsymbol{x}; \theta)$ over θ. ▷ *Step C: train π with imitation learning.*
end for
Report $\mathcal{Q} \cup \mathcal{Q}_{\mathtt{ex}}$ as the output. ▷ *Output the highly-rewarding molecules.*

Experiment (P1)

Overview of

- Comparison study of GEGL
  - Algorithms of DRL, DEO, DSL, and GA
  - $penalized \log p$ =testing ground for molecule optimization models
    - penalized octanol-water partition coefficient
  - $Penalized \log p = LogP(x) - SyntehticAccessibility(x) - RingPenalty(x)$
  - Comparison of 8,192 molecules—unrealistic results
  - $Penalized \log p$ vs $Penalized \log p$ with similarity constraints

(a) `PenalizedLogP`

Algorithm	Type	Objective
GVAE+BO [Kusner et al. 2017]	DEO	2.87 ± 0.06
SD-VAE [Dai et al. 2018]	DEO	3.50 ± 0.44
ORGAN [Guimaraes et al. 2017]	DRL	3.52 ± 0.08
VAE+CBO [Griffiths and Hernández-Lobato 2020]	DEO	4.01
ChemGE [Yoshikawa et al. 2018]	GA	4.53 ± 0.26
CVAE+BO [Gómez-Bombarelli et al. 2018]	DEO	4.85 ± 0.17
JT-VAE [Jin et al. 2018]	DEO	4.90 ± 0.33
ChemTS [Yang et al. 2017]	DRL	5.6 ± 0.5
GCPN [You et al. 2018]	DRL	7.86 ± 0.07
MRNN [Popova et al. 2019]	DRL	8.63
MolDQN [Zhou et al. 2019]	DRL	11.84
GraphAF [Shi et al. 2020]	DRL	12.23
GB-GA [Jensen 2019]	GA	15.76 ± 5.76
DA-GA [Nigam et al. 2020]	GA	20.72 ± 3.14
MSO [Winter et al. 2019]	DEO	26.1
PGFS [Gottipati et al. 2020]	DRL	27.22
GEGL† (Ours)	DRL	**31.40** ± **0.00**

(b) Similarity-constrained `PenalizedLogP`

δ	Algorithm	Type	Objective	Succ. rate
0.4	JT-VAE [Jin et al. 2018]	DEO	0.84 ± 1.45	0.84
	GCPN [You et al. 2018]	DRL	2.49 ± 1.30	1.00
	DEFactor [Assouel et al. 2018]	DEO	3.41 ± 1.67	0.86
	VJTNN [Jin et al. 2019]	DSL	3.55 ± 1.67	-
	HierG2G [Jin et al. 2020]	DSL	3.98 ± 1.09	-
	GEGL‡ (Ours)	DRL	**7.87** ± **1.81**	1.00
0.6	JT-VAE [Jin et al. 2018]	DEO	0.21 ± 0.71	0.47
	GCPN [You et al. 2018]	DRL	0.79 ± 0.63	1.00
	DEFactor [Assouel et al. 2018]	DEO	1.55 ± 1.19	0.73
	VJTNN [Jin et al. 2019]	DSL	2.33 ± 1.17	-
	HierG2G [Jin et al. 2020]	DSL	2.49 ± 1.46	-
	GEGL‡ (Ours)	DRL	**4.43** ± **1.53**	1.00

Experiment (P2)

GuacaMol Benchmark

- Designed as a benchmark performance of de novo molecular design and various tasks
- Appendix E
- Luckily, with the benchmark GEGL achieves the highest score when comparing different algorithms
- 19/20 tasks of recognizing different de Novo Molecules
- Ranolazine MPO, Sitagliptin MP, and Zaleplon MPO tasks

(a) GuacaMol

id	ChEMBL [62]	MCTS [24]	ChemGE [23]	HC-MLE [13]	GB-GA [24]	MSO [17]	CReM [26]	GEGL (Ours)
1	0.505	0.355	0.732	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
2	0.418	0.311	0.515	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
3	0.456	0.311	0.598	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
4	0.595	0.380	0.834	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
5	0.719	0.749	0.907	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
6	0.629	0.402	0.790	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
7	0.684	0.410	0.829	0.993	0.971	0.997	0.966	**1.000**
8	0.747	0.632	0.889	0.879	0.982	**1.000**	0.940	**1.000**
9	0.334	0.225	0.334	0.438	0.406	0.437	0.371	**0.455**
10	0.351	0.170	0.380	0.422	0.432	0.395	0.434	**0.437**
11	0.839	0.784	0.886	0.907	0.953	0.966	0.995	**1.000**
12	0.817	0.695	0.931	0.959	0.998	**1.000**	**1.000**	**1.000**
13	0.792	0.616	0.881	0.855	0.920	0.931	**0.969**	0.958
14	0.575	0.385	0.661	0.808	0.792	0.834	0.815	**0.882**
15	0.696	0.533	0.722	0.894	0.894	0.900	0.902	**0.924**
16	0.509	0.458	0.689	0.545	0.891	0.868	0.763	**0.922**
17	0.547	0.488	0.413	0.669	0.754	0.764	0.770	**0.834**
18	0.259	0.040	0.552	0.978	0.990	0.994	0.994	**1.000**
19	0.933	0.590	0.970	0.996	**1.000**	**1.000**	**1.000**	**1.000**
20	0.738	0.470	0.885	0.998	**1.000**	**1.000**	**1.000**	**1.000**

(b) GuacaMol with filtering

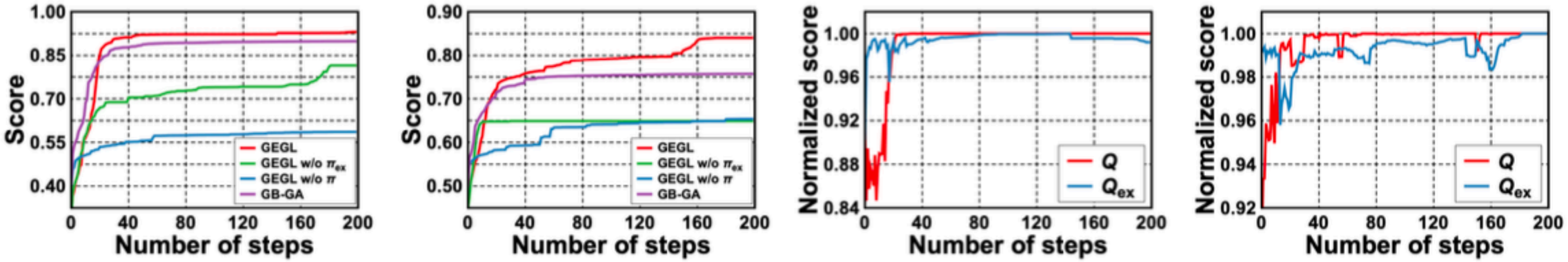
ChEMBL* [62]	ChemGE* [23]	HC-MLE* [13]	GB-GA* [24]	GEGL (Ours)
0.505	0.646	**1.000**	**1.000**	**1.000**
0.260	0.504	0.537	**0.837**	0.552
0.456	0.552	**1.000**	**1.000**	**1.000**
0.595	0.769	**1.000**	0.995	**1.000**
0.711	0.959	**1.000**	0.996	**1.000**
0.632	0.631	**1.000**	0.996	**1.000**
0.684	0.786	0.997	0.960	**1.000**
0.747	0.883	0.992	0.823	**1.000**
0.334	0.361	0.453	0.402	**0.455**
0.351	0.377	0.433	0.420	**0.437**
0.839	0.895	0.916	0.914	**1.000**
0.815	0.920	0.999	0.905	**1.000**
0.786	0.714	0.882	0.530	**0.933**
0.572	0.572	0.835	0.780	**0.833**
0.679	0.709	0.902	0.889	**0.905**
0.501	0.587	0.601	0.634	**0.749**
0.547	0.647	0.715	0.698	**0.763**
0.127	0.827	0.992	0.789	**1.000**
0.933	0.857	**1.000**	**0.994**	**1.000**
0.690	0.964	**1.000**	**1.000**	**1.000**

Experiment (P3)

Ablation studies

- Comparison study of GEGL
- Focus on max-reward priority queue
- Same types of tasks from the GuacaMl benchmark
- Compare the benchmarks of the GuacMolscores relative to the apprentice queue and the union of the two queue

$$GuacamolScore(Q)/GuacaMolScore(Q \cup Q_ex)$$



(a) Sitagliptin MPO (b) Zaleplon MPO (c) Sitagliptin MPO (d) Zaleplon MPO

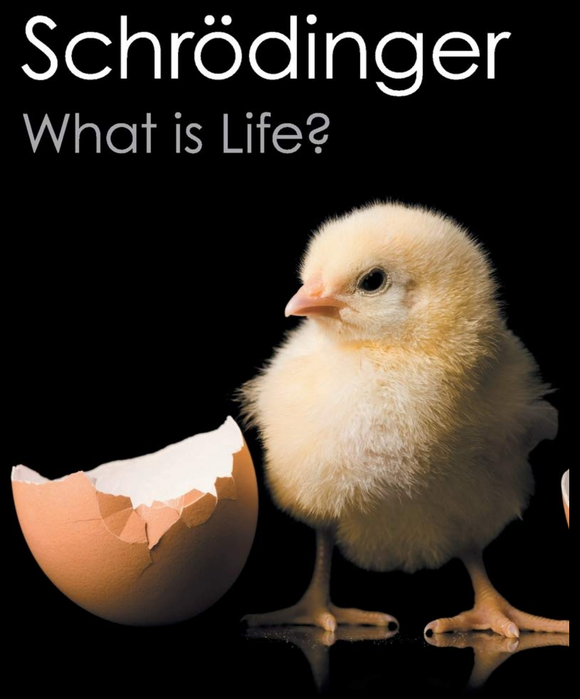
Figure 6: Illustration of ablation studies for (a, b) investigating contribution from DNN and genetic operator, and (c, d) separate evaluation of max-reward priority queues.

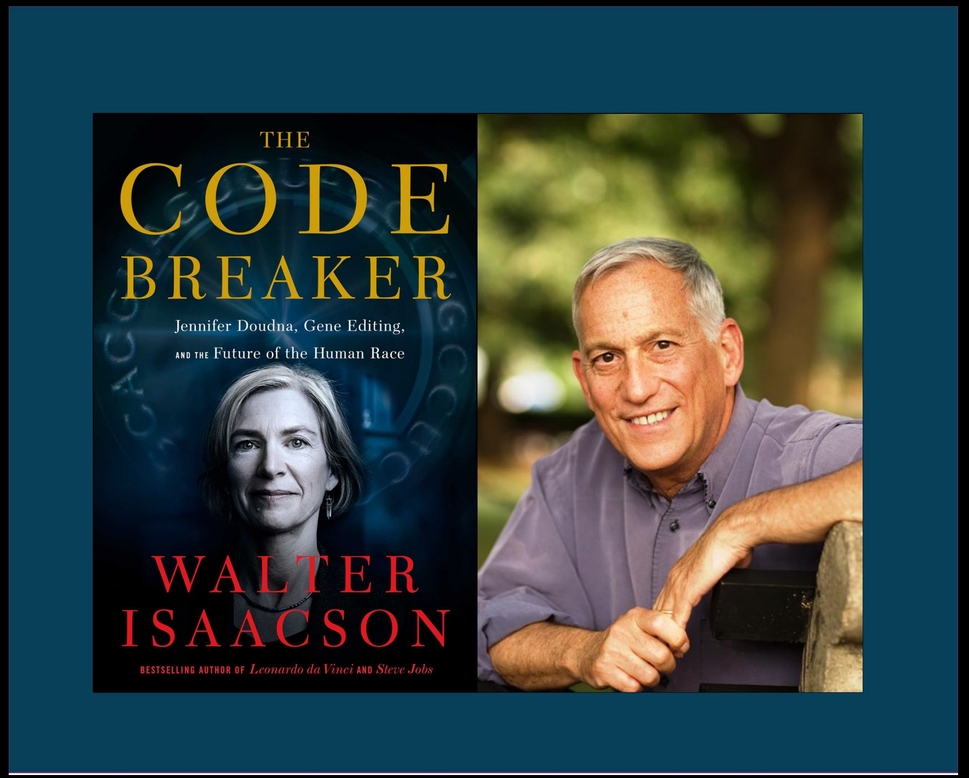
Conclusion & Broader Impacts

- The team created a new framework of DNNs to solve the design problem of molecular structures
- Optimization for a DRL framework for molecular design
- The algorithm is expected to perform well for domains of genetic algorithms
- Helping to design desire outcomes of “designer drugs”
- Relative to crossovers & Mutations

Thank you

Further Reading





Discussion

- 1. What are your thoughts on creating new drug designs?
- 2 Based on the team's efforts what are some suggestions to improve?
- 3 What are some future directions you would suggest or think of?