Supplementary Information: Chemical Design with GPU-based Ising Machines

Zetian Mao, Yoshiki Matsuda, Ryo Tamura, and Koji Tsuda

June 3, 2023

1 Discrete Latent Representation as Fingerprints

There are several reported molecule fingerprint algorithms, such as extended-connectivity fingerprint(ECFP4)[1], MinHashed fingerprint(MHFP6)[2] and atom-pair fingerprint[3], which encode molecules into bitstrings as descriptors for the target prediction. However, such mapping is not invertible, i.e., unable to translate bitstrings back to corresponding molecules. Aiming at the problem, Le et al.[4] proposed to convert the molecular fingerprint to a CDDD descriptor[5] and then decipher it using the neural network. The results show that the average reconstruction correctness is less than 70% evaluated by the Tanimoto similarity on the three test datasets because of the serious information loss during the encoding process. As a result, solving QUBO on the traditional molecule fingerprints has the defect of undecodability to valid molecules.

In contrast, we show that our fragment-based latent discrete space is well-organized. Two commonly-used drug molecules, Aspirin and Ibuprofen, are embedded to bit vectors. Figure S1(a) visualizes the intermediate molecules which are interpolated by equally flipping their different bits for 6 steps. The Tanimoto similarity to the target molecule fingerprint continuously grows as their generative fingerprints get closer. The smooth transition also implies the well-organized latent binary space. Figure S1(b) indicates the neighboring molecules around Aspirin. We first encode the Aspirin molecule to a bit vector, and then randomly flip the certain increasing number of bits. As 1 bit changes, the binary vector is still decoded to the Aspirin molecule. When 5 and 10 bits are flipped, we observed the replacement of the fragment in Aspirin. The similarity of generated molecules continuously decreases as divergence of bit vectors grows. This illustrates that molecules are mapped to a compressed distribution and those with similar fragments are in a neighbouring space.

2 Training Process

Six bJTVAE models with different latent dimensionalities, $d = 50, 100, 200, 300, 450, 600$, were trained, while the latent dimensionality of JTVAE was fixed to 56 as in

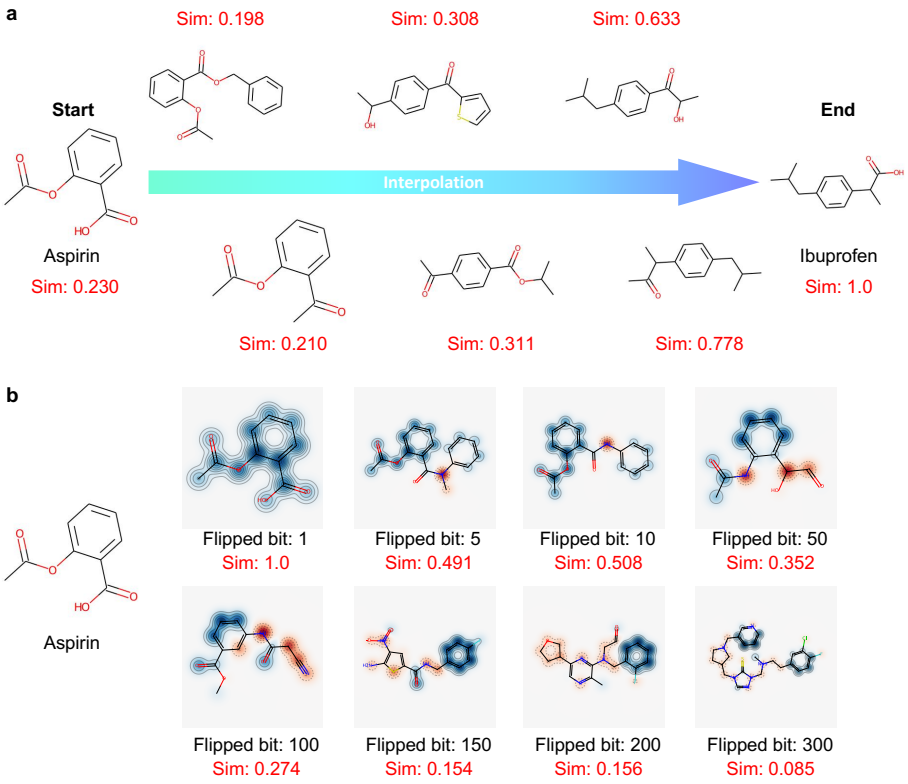

Figure S1: (a) Interpolation with equal flipped bits between two molecules in the binary space; (b) Molecular similarity varies with increasing number of flipped bits on the base molecule.

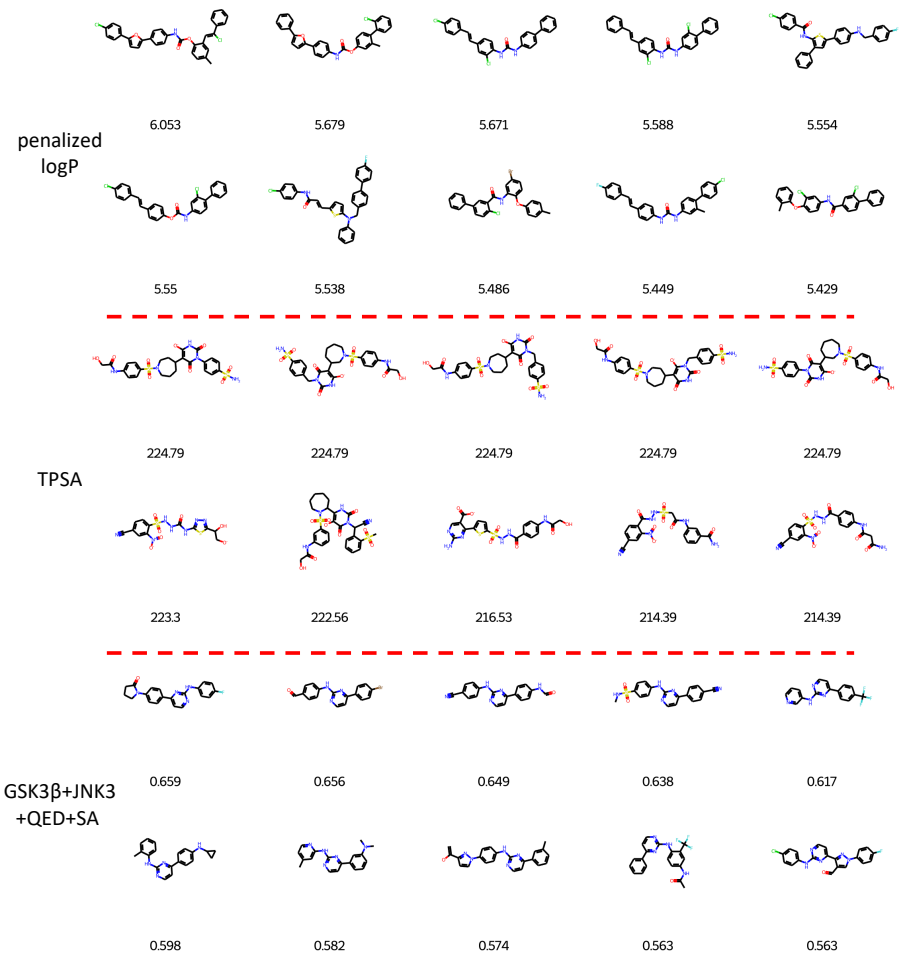

Figure S2: Top molecules generated by bVAE-IM.

Algorithm S1 bVAE-IM: Molecule generation pipeline.

Require: Define the desired number of molecule generations N

Input an *oracle* $\mathcal{O}$ for property computation
Input the unlabeled molecule dataset D_u and the labeled molecule dataset D_l with corresponding property set Y
Let $M = \emptyset$ be the set of generated molecules
Train a binary VAE model $\mathrm{M_{bVAE}}$ consisting of an Encoder and a Decoder using D_u
Map D_l to a binary vector set X with Encoder(D_l)
while $|M| < N$ **do**
    Train a factorization machine $\mathrm{M_{FM}}$ using data pairs $\{x_k, y_k\}$ for $x_k \in X$ and $y_k \in Y$
    Convert $\mathrm{M_{FM}}$ to a QUBO energy function E_{QUBO}
    Solve the global state x_i of E_{QUBO} by Ising machine
    Decode x_i to a molecule m_i with Decoder(x_i)
    Compute corresponding property y_i with $\mathcal{O}(m_i)$
    Update labeled pairs $\{X, Y\}$: $X \leftarrow X \cup \{x_i\}$, $Y \leftarrow Y \cup \{y_i\}$
    Add m_i into M: $M \leftarrow M \cup \{m_i\}$
end while
return M

literature[6]. Each model is trained with a maximum epoch of 50 on a NVIDIA Tesla V100 GPU. See Table S1 for the comparison results on training time of bJTVAE and JTVAE. It is reasonable that the training time per epoch increases as the dimensionality increment in bJTVAEs leads to more trainable parameters. Even the 600-dimensional binary representation is still more efficient to be learned than the continuous representation in terms of the average training time for each epoch.

Table S1: Comparison of training time for JTVAE and bJTVAE with different dimensional ity.

Model	Runtime per epoch	GPU device
bJTVAE-50	0.92	NVIDIA Tesla V100
bJTVAE-100	1.05	
bJTVAE-200	1.08	
bJTVAE-300	1.09	
bJTVAE-450	1.11	
bJTVAE-600	1.30	
JTVAE	1.30	

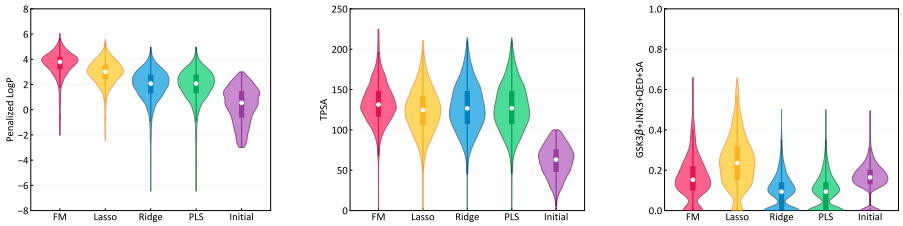

Figure S3: Property distributions of generated molecules by bVAE-IM using different surrogate models including FM, Lasso, Ridge and PLS, in the three benchmarking problems. As a reference, the property distribution in the initial labeled data is shown as well.

3 Alternative surrogate models

bVAE-IM with different regression models are applied up to the point that 300 molecules are generated. Each run is repeated five times with different random seeds. Amplify is employed as the QUBO sampler and the dimensionality of the latent space is set to 300. The binary vectors are expanded to polynomial and interaction features using the package, sklearn.preprocessing.PolynomialFeatures. Then the weight of each feature is fitted by linear models. A QUBO model can be built from the fitted weights. Figure S3 shows the integrated 5-run results. All the surrogate models show certain ability in predicting extrapolated values under our pipeline, while FM performs best among them by finding out the highest-score candidate and highest mean values on property of generated molecules in both tasks.

4 Results for Additional Properties

To prove that our method is extensively effective for diverse optimization objectives, we conducted experiments on three other properties as suggested in the GuacaMol benchmark[7]: 1) molecular weight; 2) number of aromatic rings,3) number of rotatable bonds, computed by RDKit[8]. Similarly, the training data is intentionally limited to poor properties as well: molecule weight $\in [0, 350]$, number of aromatic rings $\in [0, 2]$, number of rotatable bonds $\in [0, 5]$. Each property runs 5 times using different random seeds and 300 generations are made for each run. See Figure S4 for the integrated optimized results. In all tasks, the optimized molecules far exceed the range of training data, indicating that our approach can be well extrapolated and generalized to diverse molecule optimization problems.

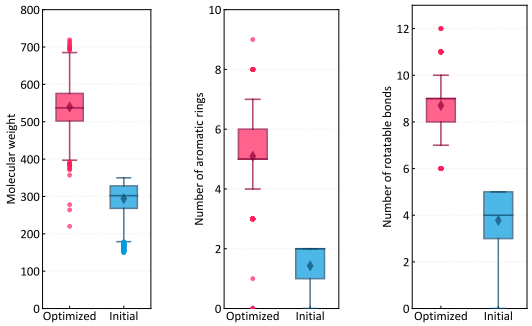

Figure S4: Comparisons of property distribution between the optimized molecule set and the initial training set for properties: 1) molecular weight; 2) number of aromatic rings; 3) number of rotatable bonds

References

[1] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[2] Daniel Probst and Jean-Louis Reymond. A probabilistic molecular fingerprint for big data settings. *Journal of cheminformatics*, 10:1–12, 2018.

[3] Raymond E Carhart, Dennis H Smith, and R Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences*, 25(2):64–73, 1985.

[4] Tuan Le, Robin Winter, Frank Noé, and Djork-Arné Clevert. Neuraldecipher–reverse-engineering extended-connectivity fingerprints (ecfps) to their molecular structures. *Chemical Science*, 11(38):10378–10389, 2020.

[5] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science*, 10(6):1692–1701, 2019.

[6] Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv:1802.04364*, 2018.

[7] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

[8] RDKit: Open-source cheminformatics. `http://www.rdkit.org`. [Online; accessed 04-March-2023].