Supplementary Information: Understanding and improving zeroth-order optimization methods on AI-driven molecule optimization

Elvin Lo*[1] Pin-Yu Chen[2]

[1]Horace Greeley High School, Chappaqua, NY 10514, USA
[2]IBM Research, Yorktown Heights, NY 10598, USA
*corresponding author(s): Elvin Lo (elvinlo922@gmail.com)

July 22, 2023

A Implementation details

Winter *et al.*[1] showed that their CDDD autoencoder model has a high validity rate of 97%, even when traversing a large distance from the valid latent representations of randomly picked molecules. In our implementation of QMO, we dealt with decode failures by assigning a penalty score of 0.1 less than the score of the starting molecule, $f(\mathbf{z}_0) - 0.1$.

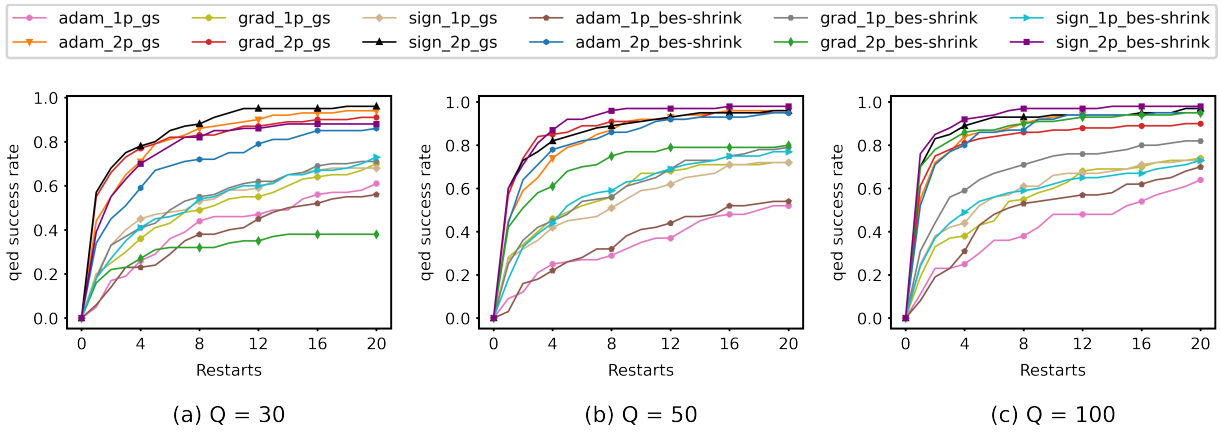
Also, we only considered the molecules generated after each optimization iteration. That is, we did not consider the Q molecules obtained from decoding the perturbed latent vectors $\{\mathbf{z} + \beta \mathbf{u}_q\}_{q=1}^{Q}$ (used for estimating gradients) in $\mathcal{Z}_{\text{iterate}}$ despite that they were also used to query the oracle $\mathcal{O}$. Especially in a realistic drug discovery scenario where oracle evaluations are highly expensive, we would of course want to also consider these molecules in case they exhibit good properties. In addition, while we considered there to be $Q + 1$ oracle evaluations necessary for each optimization iteration, the actual amount would be lower in practice as some of the perturbed latent vectors would decode to the same molecule since the perturbations are small (and a small number of latent vectors would also decode to no valid input).

All experiments were run using Google Colab, and code for QMO is available at: https://github.com/IBM/QMO-bench. For the Graph-GA and GPBO baseline models, we adopt the implementation of Gao *et al.*[2]. For Guacamol objectives, we use the implementation of the Therapeutic Data Commons (TDC)[3] (`https://tdcommons.ai`).

B Additional results

B.1 Comparing 1-point and 2-point gradient estimators

Though we ran only 2-point gradient estimators on the Guacamol tasks, we also compared 1-point gradient estimators on the QED[4] objective. Specifically, following the setup of Hoffman *et al.*[5], we defined a minimum similarity threshold of 0.4 (and did not consider molecules with similarity less than 0.4 to the starting molecule) and set the oracle $\mathcal{O}(x) = 4 \cdot \text{QED}(x) - \max(0.4 - \text{sim}(x, x_0), 0)$ for molecule $x \in \mathcal{X}$ and starting molecule x_0, where $\text{sim}(\cdot)$ denotes Tanimoto similarity with Morgan fingerprints. We selected 100 molecules with QED scores in $[0.7, 0.8]$ from the test set in Jin *et al.*[6] (who extracted the molecules from ZINC) and optimized each with $T = 20$ iterations and 20 random restarts each. We consider an optimized molecule a success if its QED scores falls in $[0.9, 1.0]$, and we visualize in Fig. B1 how many random restarts are necessary for different ZO optimization methods to achieve a given success rate. As shown, 2-point gradient estimators achieve significantly higher success rates than their 1-point counterparts given the same number of random restarts.



(a) Q = 30 (b) Q = 50 (c) Q = 100

Figure B1: Optimization of QED with different ZO optimizers.

B.2 SMILES strings of molecules found by QMO

The SMILES strings $s_1, \ldots, s_9$ from Fig. 2 and Fig. 3 of the main paper are as follows:

- s_1 = `CCCC(NC(C)Cn1c(C2CCCCC2)nc2cc(C(=O)O)ccc21)C(=O)OCC`
- s_2 = `CCCC(C(=O)OCC)c1nc2cc(C(=O)O)ccc2n1C1CCCCC1C(C)=O`
- s_3 = `CCCC(C)n1c(C(=O)NC(C)C(=O)OCC)cc2cc(C3CCCCC3)cc(C(=O)O)c21`
- s_4 = `COc1cc(C(=O)NCC23CC=CC2C3)nc2c(C#N)cccc12`
- s_5 = `CCC1(CC)C(C(=O)Nc2cccc(F)c2)=CN=C2C=C(C#N)C(=O)N21`
- s_6 = `CCC12CC(CO1)N(C(=O)c1cnc3cccc(C#N)c(=O)c3c1)C2`
- s_7 = `COc1cc2ncnc(Nc3ccc(F)c4ncccc34)c2cc1C(C)C`
- s_8 = `CCCCOc1ncccc1C(=O)CNc1ncnc2cc(OC(F)F)c(F)cc12`
- s_9 = `COc1cc(Nc2ncnc3cc(OC)c([N+](=O)[O-])cc23)ccc1F`.

B.3 Diversity metrics

Table B1 shows the diversity of the QMO optimized molecules from Section 3.2 of the main paper (optimized using sign_2p_gs with $Q = 50$). For each starting molecule in the test set, the best molecule found after two random restarts was used, showing the diversity of molecules that can be generated from different starting points. Hoffman *et al.*[5] also showed how different random restarts starting from the same starting point can find diverse candidates.

Table B1: Diversity of 20 QMO optimized molecules from different starting points.

Task	Average score	Diversity
perindopril_mpo	0.628	0.678
zaleplon_mpo	0.500	0.805
deco_hop	0.859	0.664

B.4 Query efficiency tables

Scores from Fig. 2 of the main paper are summarized below in Tables B2, B3, and B4.

Table B2: Scores for perindopril_mpo with various query budgets.

Methods	AUC	500 q	1000 q	2000 q	5000 q	10000 q
graph_ga	**0.527**	0.453	0.465	0.490	**0.533**	**0.593**
gpbo	0.502	0.446	0.478	0.478	0.494	0.583
sign_2p_gs ($Q = 30$)	0.456	0.219	0.327	0.408	0.490	0.544
sign_2p_gs ($Q = 50$)	0.441	0.176	0.284	0.388	0.485	0.541
sign_2p_gs ($Q = 100$)	0.395	0.101	0.218	0.330	0.439	0.507
graph_ga_4k + sign_2p_gs ($Q = 49$)	0.522	**0.466**	**0.491**	**0.501**	0.531	0.572
gpbo_2k + sign_2p_gs ($Q = 49$)	0.487	0.435	0.438	0.438	0.508	0.555

Table B3: Scores for zaleplon_mpo with various query budgets.

Methods	AUC	500 q	1000 q	2000 q	5000 q	10000 q
graph_ga	**0.315**	0.167	0.239	0.294	**0.337**	0.362
gpbo	0.241	0.172	0.244	0.244	0.253	0.261
sign_2p_gs ($Q = 30$)	0.259	0.001	0.013	0.105	0.321	**0.406**
sign_2p_gs ($Q = 50$)	0.233	0.000	0.007	0.048	0.291	0.390
sign_2p_gs ($Q = 100$)	0.158	0.000	0.000	0.013	0.168	0.333
graph_ga_4k + sign_2p_gs ($Q = 49$)	0.314	0.183	0.239	**0.298**	0.331	0.359
gpbo_2k + sign_2p_gs ($Q = 49$)	0.307	**0.223**	**0.254**	0.276	0.329	0.350

Table B4: Scores for deco_hop with various query budgets.

Methods	AUC	500 q	1000 q	2000 q	5000 q	10000 q
graph_ga	0.634	0.580	0.600	**0.638**	0.650	0.676
gpbo	0.663	0.587	**0.615**	0.615	0.626	0.792
sign_2p_gs ($Q = 30$)	0.582	0.529	0.539	0.572	0.638	-
sign_2p_gs ($Q = 50$)	0.652	0.529	0.548	0.576	0.669	0.783
sign_2p_gs ($Q = 100$)	0.604	0.522	0.537	0.558	0.622	0.702
graph_ga_4k + sign_2p_gs ($Q = 49$)	0.661	**0.592**	0.602	0.637	0.655	0.741
gpbo_2k + sign_2p_gs ($Q = 49$)	**0.716**	0.591	0.597	0.597	**0.772**	**0.859**

C Other ZO optimization methods

Other than the ZO optimization methods considered here, ZO stochastic coordinate descent (ZO-SCD)[7] was also tested. However, because the coordinate-wise gradient estimator relies on perturbing coordinates individually, the perturbed vector embeddings used to estimate gradients almost always decoded back to the same molecule as the original non-perturbed vector. In other words, the autoencoder almost always perceived the embedding vectors with all coordinates the same but one coordinate to be the same molecule, so the coordinate-wise gradient estimates were almost always zero.

D Hyperparameter tuning

Aside from the number of random perturbations Q, there are two other main hyperparameters for each of the ZO optimization methods compared: function smoothing parameter β, and learning rate α. The value $\beta = 10$ was used for all tasks as it was found to work well with the CDDD model. Consistent with Hoffman *et al.*[5], we find that $\beta = 1$ or below does not work well (as gradients cannot be accurately approximated without sufficient smoothing) and $\beta = 100$ or above results in many decode failures. When trying a few molecules with β values between this range (including $\beta = \{5, 10, 20, 50\}$ for each task, $\beta = 10$ still performed best for the majority of molecules. The values of α used in experiments are shown in Figure D1 while the tuning of α is shown in Table D2 and Table D3. As a note, α larger than the largest tested values for each optimization method often resulted in many decode failures, so even if the best α was the largest value tested, choosing notably larger α (greater by more than a factor of 2) may not be a good idea. Also, for ZO-Adam, two additional hyperparameters are used for the adaptive learning rate: the exponential averaging parameters β_1 and β_2. For these parameters, we use the default values used by the PyTorch Adam implementation, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Table D1: Learning rates α used for Guacamol tasks in Figure 4.

Task	Methods	$Q = 30$	$Q = 50$	$Q = 100$
perindopril_mpo	adam_2p_bes-shrink	0.2	0.3	0.3
	adam_2p_gs	0.1	0.2	0.3
	grad_2p_bes-shrink	50.0	30.0	30.0
	grad_2p_gs	2.0	2.0	5.0
	sign_2p_bes-shrink	0.1	0.1	0.1
	sign_2p_gs	0.1	0.1	0.1
zaleplon_mpo	adam_2p_bes-shrink	0.1	0.2	0.2
	adam_2p_gs	0.1	0.1	0.2
	sign_2p_bes-shrink	0.1	0.1	0.2
	sign_2p_gs	0.1	0.1	0.1
deco_hop	adam_2p_bes-shrink	0.3	0.2	0.3
	adam_2p_gs	0.3	0.3	0.3
	grad_2p_bes-shrink	70.0	70.0	70.0
	grad_2p_gs	5.0	10.0	10.0
	sign_2p_bes-shrink	0.2	0.2	0.2
	sign_2p_gs	0.1	0.2	0.2
qed	adam_1p_bes-shrink	0.2	0.2	0.2
	adam_1p_gs	0.2	0.2	0.2
	adam_2p_bes-shrink	0.2	0.2	0.2
	adam_2p_gs	0.2	0.2	0.2
	grad_1p_bes-shrink	1.5	1.5	1.5
	grad_1p_gs	0.2	0.2	0.2
	grad_2p_bes-shrink	20.0	20.0	20.0
	grad_2p_gs	1.5	2.0	2.0
	sign_1p_bes-shrink	0.2	0.2	0.2
	sign_1p_gs	0.2	0.2	0.2
	sign_2p_bes-shrink	0.2	0.2	0.2
	sign_2p_gs	0.2	0.2	0.2

References

[1] R. Winter, F. Montanari, F. Noé and D.-A. Clevert, *Chemical Science*, 2019, **10**, 1692–1701.

[2] W. Gao, T. Fu, J. Sun and C. W. Coley, Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.

[3] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. H. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun and M. Zitnik, Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), 2021.

[4] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan and A. L. Hopkins, *Nature chemistry*, 2012, **4**, 90–98.

[5] S. C. Hoffman, V. Chenthamarakshan, K. Wadhawan, P.-Y. Chen and P. Das, *Nature Machine Intelligence*, 2022, **4**, 21–31.

[6] W. Jin, K. Yang, R. Barzilay and T. Jaakkola, International Conference on Learning Representations, 2019.

[7] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang and J. Liu, Advances in Neural Information Processing Systems, 2016.

Table D2: Tuning of learning rate α for Guacamol tasks in Figure 4. Scores correspond to the average scores after optimizing 20 molecules with 2 random restarts each (40 trials total) for $T = 1000$ iterations.

Task	Methods	Learning rate α	$Q = 30$	$Q = 50$	$Q = 100$
perindopril_mpo	adam_2p_bes-shrink	0.1	0.555	0.598	0.607
		0.2	**0.578**	0.617	0.635
		0.3	0.564	**0.617**	**0.654**
	adam_2p_gs	0.1	**0.600**	0.600	0.604
		0.2	0.589	**0.611**	0.635
		0.3	0.560	0.605	**0.637**
	grad_2p_bes-shrink	30.0	0.429	**0.585**	**0.611**
		50.0	**0.466**	0.555	0.598
	grad_2p_gs	2.0	**0.584**	**0.582**	0.571
		5.0	0.500	0.566	**0.630**
	sign_2p_bes-shrink	0.05	0.531	0.593	0.602
		0.1	**0.598**	**0.630**	**0.629**
		0.2	0.531	0.575	0.615
	sign_2p_gs	0.05	0.583	0.595	0.593
		0.1	**0.585**	**0.610**	**0.635**
		0.2	0.534	0.564	0.617
zaleplon_mpo	adam_2p_bes-shrink	0.1	**0.386**	0.445	0.449
		0.2	0.208	**0.447**	**0.483**
		0.3	0.151	0.376	0.470
	adam_2p_gs	0.1	**0.455**	**0.465**	0.472
		0.2	0.374	0.453	**0.491**
		0.3	0.321	0.425	0.483
	sign_2p_bes-shrink	0.05	0.382	0.398	0.429
		0.1	**0.478**	**0.485**	0.477
		0.2	0.410	0.445	**0.485**
	sign_2p_gs	0.05	0.436	0.442	0.429
		0.1	**0.460**	**0.487**	**0.488**
		0.2	0.399	0.441	0.483
deco_hop	adam_2p_bes-shrink	0.1	0.544	0.564	0.605
		0.2	0.578	**0.636**	0.722
		0.3	**0.585**	0.628	**0.738**
	adam_2p_gs	0.1	0.564	0.585	0.603
		0.2	0.603	0.638	0.735
		0.3	**0.612**	**0.669**	**0.741**
	grad_2p_bes-shrink	50.0	0.480	0.587	0.666
		70.0	**0.508**	**0.634**	**0.739**
	grad_2p_gs	2.0	0.554	0.544	0.543
		5.0	**0.608**	0.597	0.584
		7.0	0.596	0.681	0.653
		10.0	0.592	**0.696**	**0.688**
	sign_2p_bes-shrink	0.1	0.645	0.709	0.784
		0.2	**0.663**	**0.748**	**0.860**
		0.3	0.613	0.670	0.746
	sign_2p_gs	0.1	**0.676**	0.763	0.762
		0.2	0.657	**0.783**	**0.865**
		0.3	0.616	0.621	0.763

Table D3: Tuning of learning rate α for QED task in Figure B1. Scores correspond to the average of the success rates of optimizing 100 molecules after 1, 5, and 20 random restarts with $T = 20$ iterations per restart.

Methods	Learning rate α	$Q = 30$	$Q = 50$	$Q = 100$
adam_1p_bes-shrink	0.05	0.007	0.013	0.050
	0.1	0.193	0.193	0.290
	0.2	**0.287**	**0.277**	**0.403**
adam_1p_gs	0.05	0.003	0.010	0.030
	0.1	0.180	0.200	0.237
	0.2	**0.317**	**0.290**	**0.350**
adam_2p_bes-shrink	0.05	0.187	0.363	0.497
	0.1	0.443	0.577	0.730
	0.2	**0.623**	**0.730**	**0.777**
adam_2p_gs	0.05	0.337	0.400	0.507
	0.1	0.627	0.707	0.777
	0.2	**0.723**	**0.733**	**0.787**
grad_1p_bes-shrink	0.5	0.053	0.070	0.123
	1.5	**0.440**	**0.507**	**0.590**
grad_1p_gs	0.1	0.420	0.243	0.100
	0.2	**0.437**	**0.497**	**0.453**
	0.5	0.120	0.220	0.323
grad_2p_bes-shrink	10.0	0.137	0.357	0.767
	20.0	**0.283**	**0.633**	**0.840**
grad_2p_gs	0.2	0.103	0.090	0.087
	0.5	0.350	0.323	0.260
	1.5	**0.750**	0.770	0.743
	2.0	0.697	**0.793**	**0.780**
sign_1p_bes-shrink	0.05	0.010	0.027	0.040
	0.1	0.257	0.287	0.317
	0.2	**0.453**	**0.490**	**0.503**
sign_1p_gs	0.05	0.000	0.017	0.010
	0.1	0.173	0.270	0.287
	0.2	**0.447**	**0.480**	**0.500**
sign_2p_bes-shrink	0.05	0.177	0.360	0.510
	0.1	0.497	0.723	0.810
	0.2	**0.670**	**0.833**	**0.890**
sign_2p_gs	0.05	0.293	0.373	0.483
	0.1	0.677	0.730	0.813
	0.2	**0.777**	**0.800**	**0.860**