Molecular Energy Learning Using Alternative Blackbox Matrix-Matrix Multiplication Algorithm for Exact Gaussian Process

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present an application of the blackbox matrix-matrix multiplication (BBMM) algorithm to scale up the Gaussian Process (GP) training of molecular energies in the molecular-orbital based machine learning (MOB-ML) framework. An alternative implementation of BBMM (AltBBMM) is also proposed to train more efficiently (over four-fold speedup) with the same accuracy and transferability as the original BBMM implementation. The training of MOB-ML was limited to 220 molecules, and BBMM and AltBBMM scale the training of MOB-ML up by over 30 times to 6500 molecules (more than a million pair energies). The accuracy and transferability of both algorithms are examined on the benchmark datasets of organic molecules with 7 and 13 heavy atoms. These lower-scaling implementations of the GP preserve the state-of-the-art learning efficiency in the low-data regime while extending it to the large-data regime with better accuracy than other available machine learning works on molecular energies.

1 Introduction

Machine-learning (ML) for quantum chemistry has emerged as a versatile method in chemical sciences in recent years, facilitating innovation in a variety of domains such as molecular modeling [4, 17, 20], drug discovery [6, 14, 18], and material design [9, 12, 15, 19] by delivering accurate predictions at a low computational cost. One successful example of such methods is the recently developed molecular-orbital-based machine learning (MOB-ML) [3, 13, 24], which can predict molecular energies with a high degree of accuracy while requiring relatively low amounts of training data. Nevertheless, extending MOB-ML to the large data regime has remained challenging due to the steep $O(N^3)$ complexity scaling associated with the use of Gaussian Process (GP) regression. A strategy to reduce the complexity of GP is to introduce a low-rank kernel approximation, which has been exploited in the Sparse Gaussian Process Regression [21] and Stochastic Variational Gaussian Processes [11] methods. However, such treatments of GP sometimes result in significant loss of accuracy. In contrast, Gardner et al. [7, 23] recently proposed the blackbox matrix-matrix multiplication (BBMM) method, which provides exact GP inference while reducing the training time complexity to $O(N^2)$ and allowing for multi-GPU usage.

In this work, we employ BBMM and a novel alternative implementation for BBMM (AltBBMM) to speedup and scale the GP training in MOB-ML for molecular energies. We show that AltBBMM delivers more efficient training on over 1 million pair energies without sacrificing transferability across chemical systems of different molecular sizes. The accuracy and efficiency of BBMM and AltBBMM in modeling physical problems are demonstrated by comparisons with literature results on the same datasets.

Submitted to NeurIPS 2021 AI for Science Workshop.

2 Background

2.1 Molecular orbital based machine learning (MOB-ML)

The total energy of a given chemical system can be written as the sum of the Hartree-Fock and correlation energies. The correlation energy, E_{corr}, can be further decomposed into pair energies, ϵ_{ij}, associated with occupied molecular orbitals (MO) i and j, such that $E_{\text{corr}} = \sum_{ij} \epsilon_{ij}$. MOB-ML learns pair energies by $\epsilon_{ij} \approx \epsilon^{\text{ML}}(\boldsymbol{f}_{ij})$, where the features $\boldsymbol{f}_{ij}$ describe the interactions between the molecular orbitals.[13] Due to the different scales of the values, the diagonal pair energies $\epsilon_{\text{d}} = \{\epsilon_{ij} \big| i = j\}$ and the offdiagonal pair energies $\epsilon_{\text{o}} = \{\epsilon_{ij} \big| i \neq j\}$ are trained separately, effectively producing two different models.

2.2 Gaussian Processes (GP)

Gaussian Processes (GP) are non-parametric kernel-based machine learning methods that predict the probabilistic distribution of unobserved data. Given the observed data $(X, y), X \in \mathbb{R}^{N\times d}, y \in \mathbb{R}^N$ with a Gaussian noise $\sigma^2 \in \mathbb{R}$ and a prior covariance function or kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the prediction $f(X')$ test points $X' \in \mathbb{R}^{M\times d}$ is a joint Gaussian distribution such that

$$\mathbb{E}[f(X')] = K(X', X)\hat{K}^{-1}y, \qquad \text{Var}[f(X')] = K(X', X)\hat{K}^{-1}K(X, X'), \tag{1}$$

where $\hat{K} = K(X, X) + \sigma^2 I$. The hyperparameters, which include the Gaussian noise and kernel parameters, are learned in GP training by maximizing the log marginal likelihood

$$L = -\frac{1}{2}y^T\hat{K}^{-1}y - \frac{1}{2}\log|\hat{K}| - \frac{N}{2}\log 2\pi. \tag{2}$$

The typical way to calculate the above quantities is the Cholesky decomposition, which has a time complexity of $O(N^3)$ and a memory complexity of $O(N^2)$. Such complexities limit the training size of GP-based models, such as MOB-ML, to around 50000 data points.

3 Method

3.1 Conjugate gradient (CG)

The conjugate gradient (CG) [22] algorithm offers another way to obtain the predictive mean in Eq.1 by iteratively solving $\omega = \hat{K}^{-1}y$, or equivalently $\hat{K}\omega = y$ with an $O(N^2)$ cost in each iteration. CG requires only the matrix-vector multiplications (MVMs) with the kernel matrix $\hat{K}$, which is amenable to multi-GPU acceleration. In the iteration k, the solution is found in the order-k Krylov space

$$\mathcal{K} = \text{span}\left\{\hat{K}^i y \Big| i = 0, 1, ...k-1\right\}. \tag{3}$$

The solution ω^k of CG at iteration k converges to the exact solution ω^* exponentially measured by the relative residual $\frac{||\hat{K}\omega^k - y||}{||y||}$. However, the total number of iterations k_c to converge is usually very large for a kernel $\hat{K}$ with high singularity. A common way to reduce k_c is to construct a preconditioner P and then solve the equivalent equation $P^{-1}\hat{K}\omega = P^{-1}y$ such that $P^{-1}\hat{K}$ is less singular than $\hat{K}$. [5].

Block conjugate gradient (BCG) [16], as a variant of CG, can also be used to further reduce k_c. It extends CG to solve s linear equations $\hat{K}\omega_i = y_i, i = 0, 1...s-1$ simultaneously. The number of linear equations s is also known as block size. In the iteration k of BCG, the solution is found in

$$\mathcal{K}^{\text{block}} = \text{span}\left\{\hat{K}^j y_i \Big| i = 0, 1, ...s-1, j = 0, 1, ...k-1\right\}. \tag{4}$$

By setting $y_0 = y$, and $y_i \sim N(0, I)$ for $i > 0$, BCG can converge to the same exact solution $\omega_0^* = \omega^*$ with fewer iterations since $\mathcal{K}_k \subset \mathcal{K}_k^{\text{block}}$.

3.2 Blackbox matrix-matrix multiplication (BBMM) and Alternative BBMM (AltBBMM)

BBMM [7, 23] calculates the GP inference by utilizing CG combined with the pivoted Cholesky decomposition preconditioner [1, 10]. Furthermore, a modified batched version of conjugate gradients (mBCG[1]) [7] is also proposed to estimate the marginal likelihood and its derivatives, which are required in the GP hyperparameter optimization. These enhancements reduce the training complexity to $O(N^2)$ in time, and $O(N)$ in memory and therefore enable the training of a million data points.

In this work, we propose an alternative realization of BBMM (AltBBMM) to achieve similar accuracy as BBMM with a lower cost in molecular energy prediction applications, where a low Gaussian noise ($10^{-5}\sim10^{-8}$) is required to reach the desired accuracy. However, sine the low Gaussian noise significantly increases the singularity of $\hat{K}$, CG would converge slowly or even fail to converge when the rounding errors exceed the Gaussian noise.[8] In order to speedup the CG convergence, we employ the BCG algorithm described in the previous section. The additional computational cost of BCG in each iteration is negligible compared with the kernel matrix calculations. To further improve the robustness of the convergence, we use the double-precision floating numbers in the implementation and employ the symmetric preconditioning $P^{-1/2}\hat{K}P^{-1/2}$. The Nystreoem preconditioner [5] is used as an example, but we note that better preconditioners could exist. Finally, the hyperparameters are optimized on a random subset of the entire training set in AltBBMM since the optimized hyperparameters remain similar across various training sizes for MOB-ML.

4 Computational details

We train all the models on random subsets of the QM7b-T dataset [3, 13, 24], which contains 7211 organic molecules with up to 7 heavy atoms. The test sets are the remaining QM7b-T molecules and the whole GDB-13-T dataset [3, 13, 24] containing 1000 organic molecules with 13 heavy atoms. The Matérn 5/2 kernel is used in all the GP trainings. We independently implement the BBMM according to the description of the mBCG and hyperparameter optimization in Ref. 7. The symmetric Nystroem preconditioner and the block CG are used in this work. In both BBMM and AltBBMM, the rank r of the preconditioner is chosen as 10000, the BCG block size s is fixed as 50, and the BCG iterations stop when all the s relative residuals are smaller than 10^{-6}. The hyperparameters are optimized from a full GP trained on 50 random molecules. To overcome the memory limit and maximize the multi-GPU efficiencies, the kernel computations in CG are performed in 4096×4096 batches, and such computations are dynamically distributed to all the available GPUs. Additionally, we add a Gaussian noise regularization $\sigma^2_{\text{add}} = 10^{-5}$ to the optimized Gaussian noise reduce the singularity of $\hat{K}$.

5 Results

5.1 Low noise regularization for accurate GP

We first demonstrate the necessity of utilizing a low noise regularization to achieve accurate predictions. We train all the offdiagonal energies (ϵ_{o}) pairs from 1000 QM7b-T molecules with different σ^2_{add} and test on the ϵ_{o} of the rest QM7b-T molecules. The training time and the prediction mean absolute error (MAE) are displayed in Table 1. For both BBMM and AltBBMM, regularizing with $\sigma^2_{\text{add}} = 10^{-1}$ results in a less singular $\hat{K}$ and saves half the training time, but its prediction MAE doubles when compared to the results of $\sigma^2_{\text{add}} = 10^{-5}$. Since the MOB-ML data generation is significantly more expensive than model training, we fix $\sigma^2_{\text{add}} = 10^{-5}$ for all of the following BBMM and AltBBMM experiments to achieve the most accurate model with the least amount of data.

5.2 BBMM and AltBBMM for energies of organic molecules

We now examine the accuracy and transferability of BBMM and AltBBMM in learning QM7b-T and GDB-13-T molecular energies. The transferability of MOB-ML is assessed by the MAEs per 7 heavy atoms (MAE/7HA) of test GDB-13-T molecules predicted by the models trained on QM7b-T molecules. Table 2 lists the wall-clock time of training on 6500 QM7b-T molecules by BBMM

[1] mBCG (modified batched conjugate gradients) differs from BCG (block conjugate gradient)

Table 1: Test MAEs (kcal/mol) of offdiagonal contributions ($\sum \epsilon_o$) in each molecule and training time (s) by training on ϵ_o pairs from 1000 QM7b-T molecules (N=175,795) with different Gaussian noise regularizations.

	BBMM		AltBBMM	
σ^2_{add}	Test MAE	Time	Test MAE	Time
10^{-1}	0.636	1104.30	0.619	456.46
10^{-5}	0.314	2150.93	0.312	760.54

Table 2: Test MAEs (kcal/mol) and training time (hrs) of BBMM and AltBBMM trained on 6500 QM7b-T molecules[a] with the same initial hyperparameters.

Algorithm	QM7b-T MAE	GDB-13-T MAE/7HA	Time
BBMM	0.185	0.490	26.52
AltBBMM	0.193	0.493	6.24

[a] Training size of ϵ_o is 1,152,157 and training size of ϵ_d is 124,973

and AltBBMM and the corresponding prediction MAEs on test QM7b-T and GDB-13-T molecules. Similar to the results in Table 1, by utilizing our AltBBMM approach, we gain a four-fold speedup in the training timings while only introducing 4% and 1% additional MAE in the prediction of QM7b-T and GDB-13-T, respectively, compared with BBMM.

In addition, we compare the performance of BBMM and AltBBMM with the results of the current most accurate literature methods, i.e., MOB-ML with full GP (MOB-ML/Full GP) [13] and DeePHF with extended neural network regressor (DeePHF/extended NN) [2]. The literature results of MOB-ML/Full GP are only available with up to 220 training molecules due to the limited memory resources. The introduction of BBMM and AltBBMM allows MOB-ML to scale up the training to 6500 molecules (over 1 million training pair energies) while retaining the accuracy and transferability compared with MOB-ML/full GP in Figure 1. By training on 6500 molecules, BBMM and AltBBMM reach the current best MAE/7HA for GDB-13-T as 0.490 kcal/mol and 0.493 kcal/mol, respectively. In all the cases we tested, BBMM and AltBBMM provide a better accuracy on QM7b-T and a better transferability on GDB-13-T than DeePHF/extended NN.

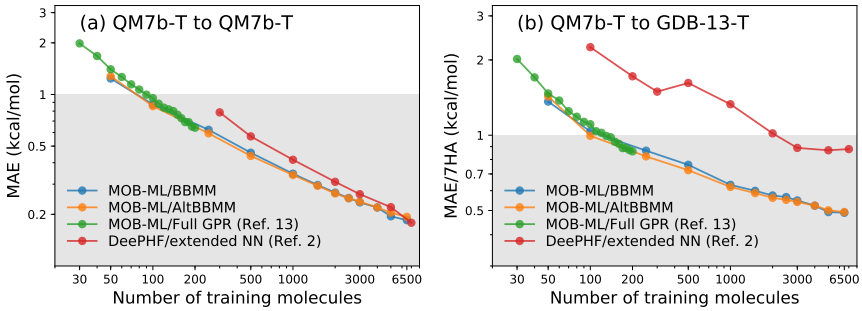

Figure 1: Learning curves for MOB-ML with different training protocols trained on QM7b-T and tested on (a) QM7b-T and (b) GDB-13-T. The accuracies of QM7b-T and GDB-13-T are measured by the MAEs and MAEs per 7 heavy atoms (MAE/7HA) of test molecules, respectively. We additionally plot the current best results in low and big data regimes, i.e., MOB-ML training with full GPR from Ref. 13 and the state-of-art DeePHF/extended NN from Ref. 2, respectively. The gray shaded area represents the chemical accuracy of 1 kcal/mol.

6 Conclusion

In this work, we successfully apply the BBMM algorithm and a new alternative implementation, AltBBMM, to Gaussian process-training for the MOB-ML method on over a million pair energies. Even though the use of BBMM alone increases our previously attainable training-set size limit over 30 times, our newly introduced AltBBMM implementation improves this further by offering a four-fold speed-up while maintaining high accuracy. With the BBMM and AltBBMM approaches, MOB-ML models can be trained using datasets with over 6500 QM7b-T molecules, yielding the best accuracy to date for the QM7b-T and GDB-13-T datasets. Future work will include extension of the approach for the prediction of other molecular properties within the MOB-ML framework.

References

[1] F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209. PMLR, 2013.

[2] Y. Chen, L. Zhang, H. Wang, and W. E. Ground state energy functional with hartreefock efficiency and chemical accuracy. *J. Phys. Chem. A*, 124(35):7155–7165, 2020.

[3] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller III. A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules. *J. Chem. Phys.*, 150 (13):131103, 2019.

[4] A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. Anatole von Lilienfeld. FCHL revisited: Faster and more accurate quantum machine learning. *J. Chem. Phys.*, 152(4):044107, 2020.

[5] K. Cutajar, M. Osborne, J. Cunningham, and M. Filippone. Preconditioning kernel matrices. In *International Conference on Machine Learning*, pages 2529–2538. PMLR, 2016.

[6] D. A. DiRocco, Y. Ji, E. C. Sherer, A. Klapars, M. Reibarkh, J. Dropinski, R. Mathew, P. Maligres, A. M. Hyde, J. Limanto, et al. A multifunctional catalyst that stereoselectively assembles prodrugs. *Science*, 356(6336):426–430, 2017.

[7] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

[8] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.

[9] S. Guerin, A. Stapleton, D. Chovan, R. Mouras, M. Gleeson, C. McKeown, M. R. Noor, C. Silien, F. M. Rhen, A. L. Kholkin, et al. Control of piezoelectricity in amino acids by supramolecular packing. *Nat. Mater.*, 17(2):180–186, 2018.

[10] H. Harbrecht, M. Peters, and R. Schneider. On the low-rank approximation by the pivoted cholesky decomposition. *Appl. Numer. Math.*, 62(4):428–440, 2012.

[11] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

[12] K. Hongo, M. A. Watson, R. S. Sanchez-Carrera, T. Iitaka, and A. Aspuru-Guzik. Failure of conventional density functionals for the prediction of molecular crystal polymorphism: a quantum monte carlo study. *J. Phys. Chem. Lett.*, 1(12):1789–1794, 2010.

[13] T. Husch, J. Sun, L. Cheng, S. J. Lee, and T. F. Miller III. Improved accuracy and transferability of molecular-orbital-based machine learning: Organics, transition-metal complexes, non-covalent interactions, and transition states. *J. Chem. Phys.*, 154(6):064108, 2021.

[14] E. E. Kwan, Y. Zeng, H. A. Besser, and E. N. Jacobsen. Concerted nucleophilic aromatic substitutions. *Nat. Chem.*, 10(9):917–923, 2018.

[15] R. J. Maurer, C. Freysoldt, A. M. Reilly, J. G. Brandenburg, O. T. Hofmann, T. Björkman, S. Lebègue, and A. Tkatchenko. Advances in density-functional calculations for materials modeling. *Annu. Rev. Mater. Res.*, 49:1–30, 2019.

[16] D. P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra Its Appl.*, 29: 293–322, 1980.

[17] Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller III. Orbnet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.*, 153(12):124111, 2020.

[18] A. R. Rosales, J. Wahlers, E. Limé, R. E. Meadows, K. W. Leslie, R. Savin, F. Bell, E. Hansen, P. Helquist, R. H. Munday, et al. Rapid virtual screening of enantioselective catalysts using catvs. *Nat. Catal.*, 2(1): 41–45, 2019.

[19] M. Rossi, P. Gasparotto, and M. Ceriotti. Anharmonic and quantum fluctuations in molecular crystals: A first-principles study of the stability of paracetamol. *Phys. Rev. Lett.*, 117(11):115702, 2016.

[20] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet–A deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, 2018.

[21] M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

[22] H. A. Van der Vorst. *Iterative Krylov methods for large linear systems*. Number 13. Cambridge University Press, 2003.

[23] K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, volume 32, pages 14648–14659. Curran Associates, Inc., 2019.

[24] M. Welborn, L. Cheng, and T. F. Miller III. Transferability in machine learning for electronic structure via the molecular orbital basis. *J. Chem. Theory Comput.*, 14(9):4772–4779, sep 2018.