Exploring Fine-Tuning Enhancements for MolCLR

Janindu Kulathunga
dept. of Computer Science and Engineering
University of Moratuwa, Sri Lanka
janindu.21@cse.mrt.ac,lk

Uthayasanker Thayasivam
dept. of Computer Science and Engineering
University of Moratuwa, Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*—MolCLR has established itself as a leading framework for self-supervised molecular representation learning, leveraging graph neural networks and contrastive learning. Its capacity to encode rich molecular features from unlabelled data makes it highly effective for downstream chemical applications. Despite its success, several standard optimisation and architectural enhancements have been proposed to further improve performance. In this work, we systematically investigate the impact of these common enhancements on the MolCLR fine-tuning process. We explore advanced optimisation strategies, including the AdamW optimiser and cosine annealing learning rate schedules, as well as architectural modifications, such as replacing global pooling with a Transformer-based readout and ensembling GIN and GCN backbones. Through systematic evaluation on established MoleculeNet benchmarks, we find that these modifications do not yield significant or consistent performance gains over the well-tuned baseline MolCLR framework. This highlights the inherent robustness of the pre-trained representations but also suggests that future performance gains may require more fundamental changes to the pre-training objectives or data augmentation strategies, rather than standard fine-tuning adjustments.**

***Index Terms*—MolCLR, Molecular Representation Learning, Self-Supervised Learning, Graph Neural Networks (GNNs), Fine-Tuning, Model Robustness, Optimisation Strategies, AdamW, Graph Transformer, Ensemble Methods, Cheminformatics, MoleculeNet**

I. Introduction

Self-supervised molecular representation learning has increasingly become a cornerstone in modern drug discovery and computational chemistry, primarily due to the limited availability of high-quality labelled datasets and the high cost associated with experimental data generation. Accurate molecular embeddings enable models to generalise across a wide range of chemical tasks, from predicting physicochemical properties to assessing bioactivity and toxicity. MolCLR [2] represents a significant advancement in this domain, employing contrastive learning in combination with graph neural networks (GNNs) to learn rich, transferable molecular representations from unlabelled molecular graphs. These embeddings can then be fine-tuned for diverse downstream tasks, offering a practical solution to the challenge of data scarcity.

Despite its effectiveness, it is often assumed that the MolCLR framework can be readily improved by applying modern, standard enhancements from the wider deep learning field. This paper presents a systematic investigation into the practical impact of these common enhancements. We explore advanced optimisation techniques such as AdamW with cosine annealing, architectural modifications including graph transformer readouts, and ensemble models. The objective of this work is to quantify the impact of these strategies and determine their efficacy in the context of fine-tuning pre-trained molecular models. Through extensive evaluation on MoleculeNet benchmarks, we demonstrate that these incremental improvements, while theoretically promising, do not yield significant performance gains over the robust and well-tuned MolCLR baseline. This result underscores the stability of the original framework but also suggests that future performance gains are not easily achieved through standard fine-tuning modifications, motivating a deeper look into pre-training objectives and augmentation strategies. Modifications tried can be viewed here.

II. Related Works

MolCLR has emerged as a prominent framework for self-supervised molecular representation learning, combining GIN-based graph neural network (GNN) encoders with a variety of graph-level augmentations to facilitate contrastive pre-training [2]. Specifically, MolCLR employs atom masking, bond deletion, and subgraph removal to generate multiple views of the same molecule, thereby enabling the model to learn robust embeddings that capture both local atomic environments and the global topological structure of molecules. These embeddings are highly transferable across a range of downstream molecular property prediction tasks, including classification tasks such as chemical toxicity (e.g., BBBP, Tox21) and regression tasks such as solubility or free energy prediction. By leveraging large-scale unlabelled molecular databases containing millions of compounds, MolCLR is able to learn representations that remain informative even when fine-tuned on smaller, labelled datasets, demonstrating superior performance in low-resource scenarios.

The foundational aspect of MolCLR builds upon prior advancements in molecular representation techniques. Traditional approaches, such as Extended-Connectivity Fingerprints (ECFP) [3], encode circular atom neighbourhoods into fixed-length binary vectors, while linear representations like SMILES [4] and SELFIES [5] facilitate the application of sequence-based models, including recurrent neural networks and transformers. Although effective in capturing local chemical information, these methods often fail to encode the full graph structure of molecules, particularly three-dimensional conformations and stereochemistry, which can limit generalisation to unseen compounds. Graph Neural Net-

works (GNNs) address this limitation by treating molecules as graph-structured data, with atoms represented as nodes and bonds as edges. Variants such as Graph Convolutional Networks (GCNs) [6] and Graph Isomorphism Networks (GINs) [7] propagate and aggregate information across neighbouring nodes, with GINs demonstrating particularly strong discriminative capabilities. Quantum-aware architectures, including SchNet [8], integrate continuous atomic coordinates to model interactions, while Message Passing Neural Networks (MPNNs) [9] generalise this further by incorporating rich edge features such as bond type, distance, and angular information, allowing accurate modelling of chemical interactions.

Self-supervised learning approaches have further extended the capabilities of molecular representation learning. Sequence-based transformers, such as ChemBERTa [10] and SMILES-BERT [11], learn contextual embeddings from large SMILES corpora, while graph-based methods, including those proposed by Hu et al. [12] and the N-Gram Graph approach [13], exploit node- and graph-level pretext tasks or subgraph co-occurrence statistics to learn embeddings without requiring labelled data. More recently, contrastive learning frameworks for graphs, as introduced by You et al. [14], maximise agreement between augmented views of the same graph while minimising agreement with different graphs, providing a powerful mechanism to learn discriminative and transferable representations. MolCLR extends these ideas to the molecular domain, carefully designing augmentations that respect chemical validity and generate meaningful positive and negative pairs for contrastive pre-training.

Despite its strong performance, MolCLR's framework can benefit from several enhancements. Prior studies have demonstrated that optimisers such as AdamW [18] can improve generalisation by decoupling weight decay from the learning rate, while advanced learning rate schedules such as cosine annealing with warm restarts [17] can facilitate smoother convergence and escape from local minima. Architectural improvements, including graph transformer blocks [19] and motif-based attention mechanisms [20], have been shown to capture long-range dependencies and emphasise chemically salient substructures, improving interpretability. Furthermore, the application of chemically informed data augmentations [21] that preserve functional groups, stereochemistry, and other chemical semantics enhances robustness and transferability of the embeddings. Together, these developments suggest a clear pathway for systematically enhancing MolCLR, resulting in more expressive, interpretable, and generalisable molecular representations suitable for a wide range of cheminformatics applications.

III. Enhancement Methods

A. Optimisation and Training Strategies

To enhance fine-tuning stability and model generalisation, several key adjustments are made to the training paradigm. The conventional Adam optimiser is replaced by AdamW, which decouples weight decay from the gradient update, providing more effective regularisation and preventing overfitting. A crucial strategy introduced is differential learning rates. The pre-trained GNN backbone (base) is fine-tuned with a low learning rate, while the newly-initialised prediction head is trained with a significantly higher learning rate. This approach preserves the robust, pre-trained representations in the base layers while allowing the task-specific head to adapt quickly. This is complemented by a cosine annealing learning rate schedule (with or without warm restarts) to dynamically modulate the learning rate, helping the model settle into broad, generalisable minima. Finally, early stopping is implemented as a primary regularisation measure. The model's performance on a held-out validation set is monitored, and training is terminated if the validation metric (e.g., ROC-AUC or MAE) fails to improve for a specified number of epochs. This prevents the model from overfitting to the training data and ensures the saved checkpoint represents the point of peak generalisation.

B. Architecture Modifications

The model's architecture is significantly enhanced to improve representational power. First, the standard global pooling layer (e.g., global mean/max/add) is replaced with a Transformer-based readout mechanism. This module functions as a sophisticated pooling layer by appending a learnable [CLS] token to the set of node embeddings and processing the entire sequence through a Transformer encoder. The resulting embedding for the [CLS] token, which captures complex global interactions between all nodes, serves as the final graph-level representation. Furthermore, an ensemble model is adopted to leverage the complementary strengths of different GNN architectures. The final model utilises two distinct, pre-trained backbones: a Graph Isomorphism Network (GIN) and a Graph Convolutional Network (GCN). The graph-level representations from both models are concatenated and passed to a final ensemble pred head, allowing the model to integrate diverse structural and feature-based perspectives for the downstream task.

IV. Results

A. Hyperparameter Tuning

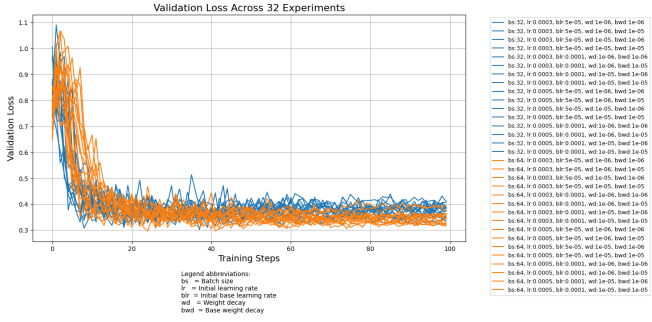

Fig. 1. Validation loss curves across 32 experiments. Legend shows batch size (bs), initial learning rate (lr), base learning rate (blr), weight decay (wd), and base weight decay (bwd).

For the 32 fine-tuning experiments on the BBBP dataset, we explored variations in several hyperparameters. The search space for each hyperparameter is shown below:

- **Batch size (bs)**: 32, 64
- **Initial learning rate (lr)**: 0.0003, 0.0005
- **Initial base learning rate (blr)**: 0.00005, 0.0001
- **Weight decay (wd)**: 1e-6, 1e-5
- **Base weight decay (bwd)**: 1e-6, 1e-5
- **Model type**: gin
- **Number of GNN layers**: 5
- **Feature dimension**: 512
- **Dropout ratio**: 0.3
- **Readout pooling method**: mean

These hyperparameters were systematically combined to generate all 32 experimental settings. Each experiment was fine-tuned from a pre-trained GIN model (*fine_tune_from* = *pretrained_gin*) and evaluated on the validation set after each epoch.

B. Different Base Learning Rate and Weight Decay

TABLE I
EFFECT OF BASE WEIGHT DECAY ON TEST PERFORMANCE

Base Weight Decay (10^x)	Test Loss	Test ROC AUC
10^{-6}	1.4743	0.7323
10^{-4}	1.1772	0.7023
10^{-8}	1.4455	0.7417

In the initial experiments, a common weight decay (10^{-6}) was applied to both the GNN base encoder and the prediction head. To investigate the effect of decoupling regularisation, we experimented with different base weight decay (*bwd*) values of 10^{-4} and 10^{-8}, while keeping the head weight decay constant.

Table shows the loss,ROC-AUC across these runs. The results indicate that varying the base weight decay has only a minor impact on the overall validation loss, similar to the effect observed when adjusting the base learning rate (*blr*). This suggests that, for the current model and dataset, the training dynamics are relatively insensitive to moderate changes in base regularisation. Nevertheless, slight improvements are observed in some runs, highlighting that careful tuning of base weight decay may still be beneficial for stabilising training or preventing overfitting in certain configurations.

C. Optimiser Changing

The initial experiments were conducted using the Adam optimiser, which applies standard weight decay uniformly across both the GNN base encoder and the prediction head. To investigate whether a different optimisation strategy could improve convergence or generalisation, we additionally experimented with AdamW, which decouples weight decay from the learning rate and has been shown in some contexts to enhance performance and stability.

Figure 2 illustrates the validation loss curves for runs using both Adam and AdamW optimisers. While minor fluctuations in validation loss are observed between the two settings,

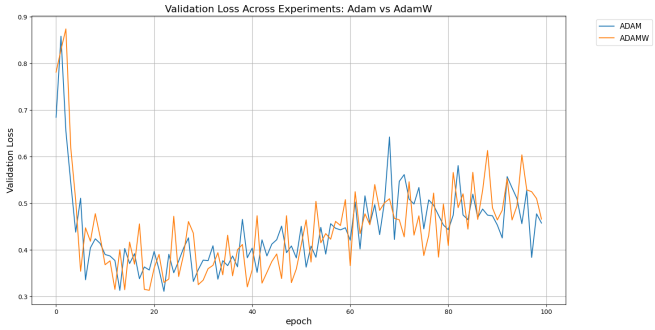

Fig. 2. Validation loss curves comparing the Adam and AdamW optimisers.

there is no clear or consistent improvement attributable to switching the optimiser. This suggests that, for the current model architecture, dataset, and hyperparameter range, the choice between Adam and AdamW does not significantly affect downstream performance. Nevertheless, AdamW may still provide benefits in terms of training stability or longer-term generalisation in other tasks or with larger models, and its effect might be more pronounced when combined with more aggressive learning rate schedules or regularisation schemes.

D. Scheduler Using

To explore the potential benefits of dynamic learning rate adjustment, we conducted additional experiments incorporating learning rate schedulers. Specifically, we tested both the Cosine Annealing LR and Cosine Annealing with Warm Restarts schedules, which are designed to modulate the learning rate during training to potentially improve convergence and avoid local minima.

Using these schedulers did not lead to substantial improvements in validation loss relative to the baseline setting with no scheduler. Across both classification and regression tasks, the curves remain largely similar, suggesting that, for the current model architecture, dataset, and hyperparameter ranges, the additional complexity introduced by learning rate scheduling does not significantly affect training dynamics or model performance.

It is possible that the lack of improvement arises from the relatively modest dataset size or the robustness of the chosen base learning rates. Future work could explore more aggressive scheduling parameters or combine schedulers with other optimisations, such as AdamW or modified weight decay, to evaluate potential synergistic effects.

E. Transformer

To enhance the model's ability to capture global molecular properties and long-range atomic dependencies, we replaced the standard global pooling layer (e.g., global mean/max/add) with a Transformer-based readout mechanism. This `GraphTransformerPool` module appends a learnable `[CLS]` token to the sequence of node embeddings generated by the GNN backbone. The entire set of tokens is then processed by a Transformer encoder, and the final embedding

of the [CLS] token is used as the graph-level representation, which is then passed to the downstream prediction head.

Despite the theoretical increase in expressive power, this architectural modification did not result in a consistent improvement in performance. Validation metrics across both classification and regression tasks remained comparable to, but not significantly better than, the baseline models using simple global-add pooling. This suggests that the additional parameters introduced by the Transformer, which are trained from scratch during fine-tuning, may lead to overfitting on the smaller downstream datasets. It is also possible that the global information captured by the 5-layer GNN backbone is already sufficient for the predictive tasks studied.

F. Ensemble Model

To leverage the complementary representational strengths of different GNN architectures, we implemented an ensemble model. This model processes input data through two parallel, pre-trained backbones: a Graph Isomorphism Network (GIN) and a Graph Convolutional Network (GCN). The graph-level feature vectors from each model (after their respective `feat_lin` layers) are then concatenated to create a single, combined representation. This larger vector is fed into a new, dedicated `ensemble_pred_head` that is trained from scratch.

Similar to the Transformer readout, this feature-level ensemble also failed to produce a significant performance increase. The ensemble's performance was generally on par with the stronger of the two individual models (GIN or GCN), but it did not show a clear synergistic effect. This outcome may indicate that the features learned by the GIN and GCN backbones are already highly correlated after the MolCLR pre-training. Alternatively, the increased dimensionality of the concatenated feature vector and the new prediction head may require more data or stronger regularization to train effectively without overfitting.

G. Summary

We conducted a systematic evaluation of fine-tuning strategies for the pre-trained MolCLR model. A 32-experiment grid search on hyperparameters (Figure 1) showed that performance is relatively insensitive to modest changes in learning rates, weight decay, and batch size. Furthermore, advanced optimisation techniques, including using the AdamW optimiser (Figure 2), decoupling base weight decay (Table I), and applying cosine annealing schedulers, failed to produce significant gains. Major architectural modifications, such as replacing the standard global pooling with a Transformer-based readout and ensembling GIN and GCN models, also did not yield performance enhancements. Overall, the pre-trained model proves robust but highly resistant to improvement from these common fine-tuning strategies, suggesting more fundamental modifications may be required to achieve substantial gains.
Overall, these preliminary results indicate that simple modifications of individual hyperparameters do not lead to substantial gains, motivating more sophisticated enhancements in architecture, augmentation, or combined training strategies.

V. Discussion

The experiments conducted to enhance MolCLR provide several key insights. The consistent finding across all tested modifications is the remarkable robustness of the baseline MolCLR framework. Neither fine-grained hyperparameter tuning, advanced optimiser strategies, nor significant architectural modifications could substantially improve upon the performance of the original, well-tuned model.

A. Hyperparameter Sensitivity

The experiments varying the base learning rate (*blr*) and base weight decay (*bwd*) showed only minor differences in validation performance (Figure 1, Table I). This suggests that, within the tested ranges, the model is relatively insensitive to these adjustments. The pre-trained GNN weights appear to be well-regularised and stable, not benefiting significantly from decoupled weight decay or minor learning rate changes.

B. Optimisers and Schedulers

Experiments comparing Adam and AdamW (Figure 2) showed no clear advantage for either optimiser. Similarly, incorporating Cosine Annealing schedules did not lead to noticeable improvements. The training dynamics of the fine-tuning process appear stable enough that these advanced optimisation strategies, which are often beneficial in training from scratch, provide no tangible benefit here.

C. Architectural Modifications

Our investigation extended to significant architectural changes, which also failed to improve performance. Replacing the standard global pooling with a `GraphTransformerPool` did not yield performance gains. This suggests that the additional parameters, which are trained from scratch, may struggle to generalise on smaller downstream datasets or that the global information captured by the 5-layer GNN backbone is already sufficient. Likewise, the feature-level ensemble of GIN and GCN models failed to show a clear synergistic effect. This may indicate that the representations learned by the two backbones are already highly correlated after MolCLR pre-training, offering little new information when combined.

D. Overall Observations

These results highlight that the pre-trained MolCLR representations are highly transferable but also highly resistant to improvement via common fine-tuning strategies. The performance bottleneck does not seem to lie in suboptimal fine-tuning hyperparameters or a simplistic readout function.

VI. Conclusion

In this work, we systematically evaluated a suite of common enhancement strategies for fine-tuning the MolCLR framework. These included a 32-experiment grid search on hyperparameters, the adoption of the AdamW optimiser, the use of cosine annealing learning rate schedulers, the replacement of global pooling with a Transformer readout, and the implementation of a GIN/GCN ensemble model.

Our comprehensive experiments conclusively demonstrate that none of these modifications yield significant or consistent performance improvements over the well-tuned baseline. These findings highlight two key insights: first, the pre-trained MolCLR framework is exceptionally robust and its representations are stable across a range of fine-tuning configurations. Second, substantial performance gains are not easily achieved through standard optimisation or architectural adjustments at the fine-tuning stage. Future work, therefore, should pivot away from these common techniques and instead explore more fundamental modifications, such as the design of novel chemically-informed data augmentations, adjustments to the contrastive pre-training objective itself, or leveraging larger and more diverse pre-training datasets.

References

[1] L. David, A. Thakkar, R. Mercado, and O. Engkvist, "Molecular representations in AI-driven drug discovery: a review and practical guide," *J. Cheminformatics*, vol. 12, no. 1, pp. 1–22, 2020.

[2] Y. Wang, J. Wang, Z. Cao, and A. B. Farimani, "Molecular contrastive learning of representations via graph neural networks," *arXiv preprint arXiv:2102.10056*, 2021.

[3] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, 2010.

[4] D. Weininger, "SMILES, a chemical language and information system," *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, 1988.

[5] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "SELFIES: A 100% robust molecular string representation," *Mach. Learn.: Sci. Technol.*, vol. 1, no. 4, p. 045024, 2020.

[6] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[7] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. ICLR*, 2019.

[8] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K. R. Müller, "SchNet: A deep learning architecture for molecules and materials," *J. Chem. Phys.*, vol. 148, no. 24, p. 241722, 2018.

[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. ICML*, 2017.

[10] S. Chithrananda, G. Grand, and B. Ramsundar, "ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction," *arXiv preprint arXiv:2010.09885*, 2020.

[11] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "SMILES-BERT: Large scale unsupervised pre-training for molecular property prediction," in *Proc. 10th ACM Conf. Bioinf., Comput. Biol., Health Informat. (BCB)*, 2019.

[12] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. L. Irwin, P. F. Riley, and J. Leskovec, "Strategies for pre-training graph neural networks," in *Proc. ICLR*, 2020.

[13] S. Liu, M. F. Demirel, and Y. Liang, "N-gram graph: Simple unsupervised representation for graphs," in *Proc. NeurIPS*, 2019.

[14] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. NeurIPS*, 2020.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020.

[16] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. NeurIPS*, 2016.

[17] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[19] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proc. NeurIPS*, 2019.

[20] W. Jin, R. Barzilay, and T. Jaakkola, "Hierarchical generation of molecular graphs using structural motifs," in *Proc. ICML*, 2020.

[21] K. Do, T. Tran, and S. Venkatesh, "Graph transformation policy network for chemical reaction prediction," in *Proc. KDD*, 2019.

[22] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.