TRANSFER LEARNING WITH DIFFUSION MODEL FOR POLYMER PROPERTY PREDICTION

Gang Liu
Department of Computer Science and Engineering
University of Notre Dame
gliu7@nd.edu

Meng Jiang
Department of Computer Science and Engineering
University of Notre Dame
mjiang2@nd.edu

ABSTRACT

Polymers are important and numerous. While the structure synthesis and property annotation for polymers require expensive equipment and a long time of effort, small molecules without annotations have been collected from various sources and at a large scale. However, there is a lack of studies for effective transfer learning from molecules without labels (as the source domain) to polymers with labels (as the target domain). This paper proposes to extract the knowledge underlying the large set of source molecules as a specific set of useful graphs to augment the training set for target polymers. We learn a diffusion probabilistic model on the source data and design two new objectives to guide the model's denoising process with target data to generate target-specific labeled graphs. Experiments from unlabeled molecules to labeled polymers demonstrate that our transfer learning approach outperforms existing semi/self-supervised learning approaches.

1 INTRODUCTION

Polymers in materials are macromolecules, composed of many repeating units. Their attractive properties are broadly applied to plastic cups, aerospace structures, etc. Given a graph structure in the repeating unit, the accurate prediction of the polymer property is important to the novel material discovery. However, both polymer synthesis and annotation require specialized knowledge, as well as lengthy and costly experiments in wet labs Hsissou et al. (2021). It requires us to utilize the large set of molecules at hand to transfer useful knowledge to the target of polymer property prediction.

However, annotating numerous chemical properties for molecules suffers similar time and cost issues as annotating polymer properties. It challenges the transfer learning from source molecules to target polymers because the source is unlabeled. Existing semi/self-supervised learning usually cannot effectively transfer knowledge across two domains. Semi-supervised learning like self-training improperly assigns polymer properties to small molecules and may overlook useful knowledge contained in molecules with low prediction confidence. Self-supervised learning leverages knowledge from all source molecules. But the knowledge extracted from hand-crafted tasks in self-supervised learning may conflict with the knowledge required in the target polymer domain. Aromatic rings, for instance, are a prevalent structure in molecules Maziarka et al. (2020) and are considered valuable in self-supervised tasks Zhang et al. (2021). However, polymer properties such as oxygen permeability can be more related to non-aromatic rings in some cases Liu et al. (2022), which would be overlooked if not using tailored self-supervised tasks specifically for the target task. As self-supervised tasks strive for universality across various targets, the transferred knowledge may force the property predictor in the target domain to focus more on aromatic rings, leading to poor prediction.

To address the above problems, we propose a *Data-Centric Transfer* learning framework (DCT). It extracts knowledge from all source molecules and avoids the use of self-supervised learning that has inappropriate hand-crafted tasks. We use a diffusion probabilistic model (known as *diffusion model*) to capture the data distribution of source data, leveraging its capability of distribution coverage, stationarity, and scalability Dhariwal & Nichol (2021). At the stage of performing a particular target task for polymer property prediction, the reverse process in the diffusion model, guided by two novel target-related optimization objectives, generates new target-specific labeled examples. Minimal suf-

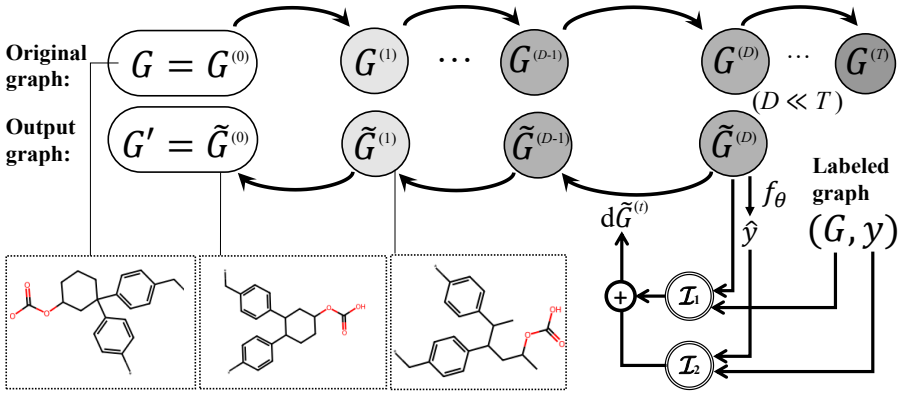

Figure 1: The diffusion model in DCT: It performs target-specific data augmentation using two target-related optimization objectives $\mathcal{I}_1$ and $\mathcal{I}_2$ in the reverse process. The model was trained on source graphs to learn the general data distribution. Then it generates $(G', y' = y)$ based on the graphs (G, y) from the target domain in the reverse process. It perturbs G with D steps and optimizes G' to be minimally similar to G (Objective $\mathcal{I}_1$) and sufficiently preserve the label of G (Objective $\mathcal{I}_2$).

ficient knowledge from the source molecules is transferred into these augmented examples, and then to enhance the training of prediction models in the target domain.

2 Preliminary

Given a target task for polymer property prediction, there are N labeled graphs: $\{(G_i, y_i)\}_{i=1}^N$. The prediction model f consists of a GNN and a multi-layer perceptron (MLP). We consider Graph Isomorphism Networks (GIN) Xu et al. (2019) to encode graph structures. Given a polymer graph G (*e.g,* in Figure 1), GIN updates the representation vector of node v in the graph at l-layer as follows.

$$\mathbf{h}_v^l = \mathrm{MLP}^l\left((1+\epsilon)\cdot \mathbf{h}_v^{l-1} + \sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{l-1}\right), \tag{1}$$

where ϵ is a learnable scalar and $u \in \mathcal{N}(v)$ is one of node v's neighbor nodes. After stacking L layers, we could get the predicted label $\hat{y}$ after a READOUT function (*e.g.,* summation) on the graph:

$$\hat{y} = \mathrm{MLP}\left(\mathrm{READOUT}\left(\left\{\mathbf{h}_v^L \mid v \in G\right\}\right)\right). \tag{2}$$

However, f is hardly well-trained since the labeled polymers are limited. Fortunately, we have a large number of unlabeled molecules that can be utilized as the source data for transfer learning.

3 DCT: A Transfer Learning Framework with Diffusion Model

3.1 Learning Data Distribution from Source Domain

As in Figure 1, the diffusion model corrupts the source graphs (molecules) to a standard normal distribution by slowly perturbing the data with noise in T steps. Then the model learns the time-dependent gradient field of the perturbed data distribution to generate graphs from noise. Given a graph G from the source domain, we use continuous time $t \in [0, T]$ to index multiple diffusion steps $\{G^{(t)}\}_{t=1}^T$, such that $G^{(0)}$ follows the original data distribution and $G^{(T)}$ follows a prior distribution like the normal distribution. The forward diffusion is a stochastic differential equation (SDE):

$$\mathrm{d}G^{(t)} = \mathbf{f}\left(G^{(t)}, t\right)\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}, \tag{3}$$

where $\mathbf{w}$ is the standard Wiener process (Brownian motion), $\mathbf{f}(\cdot, t) : \mathcal{G} \to \mathcal{G}$ is the drift coefficient and $g(t) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient. $\mathbf{f}(G^{(t)}, t)$ and $g(t)$ relate to the amount of noise added

to the graph at each infinitesimal step t. The reverse-time SDE uses scores of the perturbed graphs, or $\nabla_{G^{(t)}} \log p_t(G^{(t)})$, for denoising and graph generation from T to 0 Song et al. (2021):

$$\mathrm{d}G^{(t)} = \left[\mathbf{f}(G^{(t)}, t) - g(t)^2 \nabla_{G^{(t)}} \log p_t(G^{(t)})\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{4}$$

where $p_t(G^{(t)})$ is the marginal distribution at time t in forward diffusion. $\overline{\mathbf{w}}$ is a reverse time standard Wiener process. $\mathrm{d}t$ here is an infinitesimal negative time step. The score $\nabla_{G^{(t)}} \log p_t(G^{(t)})$ is unknown in practice and it is approximated by the score function $\mathbf{s}(G^{(t)}, t)$ with score matching techniques Song et al. (2021). On graphs, Jo et al. (2022) used two GNNs to develop the score function $\mathbf{s}(G^{(t)}, t)$ to de-noise both node features and graph structures.

3.2 Generating Labeled Graphs in Target Domain

Given a labeled polymer graph (G, y) from the target domain, the new labeled graph (G', y') is expected to provide *useful knowledge to augment* the training set in the target domain. We name it *the augmented graph* throughout this section. The augmented graph is desired to have two properties: (1) Target relatedness and (2) Diversity. *Target relatedness* indicates that (G', y') are from the graph/label spaces of the target domain where (G, y) come from and thus transfer *sufficient target knowledge* into the training set. *Diversity* indicates the augmentation aims to learn from source to create diverse target data points, which should contain *minimal target knowledge* about G. Otherwise, G' would cause severe over-fitting issues in the target domain if G' was too similar to G.

Both optimizations of the augmented graphs G' can be formulated using mutual information $\mathcal{I}(\cdot\,;\cdot)$:

Definition 3.1 (Sufficiency for Data Augmentation)**.** The augmented graph G' sufficiently preserves the target label of the original graph G if and only if $\mathcal{I}(G'; y) = \mathcal{I}(G; y)$.

Definition 3.2 (Minimal Sufficiency for Data Augmentation)**.** The Sufficiency is minimal for data augmentation if and only if $\mathcal{I}(G'; G) \leq \mathcal{I}(\bar{G}; G)$, $\forall \bar{G}$ is sufficient.

Specifically, the augmented graph G' could be optimized using any (G, y) from the target domain:

$$\min_{\mathcal{I}_1} \max_{\mathcal{I}_2} \; \mathbb{E}_G \left[\mathcal{I}_1\left(G'; G\right) + \mathcal{I}_2\left(G'; y\right)\right]. \tag{5}$$

For the first objective, we use the leave-one-out variant of InfoNCE Poole et al. (2019); Oord et al. (2018) as the upper bound estimation. For the i-th labeled graph (G_i, y_i),

$$\mathcal{I}_1 \leq \mathcal{I}_{\text{bound}}(G_i'; G_i) = \log \frac{p(G_i'|G_i)}{\sum_{j=1, j\neq i}^{M} p(G_i'|G_j)}, \tag{6}$$

where G_i' is the augmented graph. When G_i' is optimized, G_i makes a positive pair; $\{G_j\}$ $(j \neq i)$ are $M-1$ negative samples of labels that do not equal y_i. (M is a hyperparameter.) We use cosine similarity and a softmax function to calculate $p(G_i'|G_j) = \frac{\exp(\text{sim}(G_i', G_j))}{\sum_{j=1}^{M} \exp(\text{sim}(G_i', G_j))}$. In practice, we extract statistical features of graphs to calculate their similarity. Details are in appendix B.2.

For the second objective, we denote the predicted label of the augmented graph G' by $f(G')$. We maximize the log likelihood $\log p\left(y|f(G')\right)$ to maximize $\mathcal{I}_2(G'; y)$. Specifically, after the predictor f is trained for several epochs in the target domain, we freeze its parameters and use it to optimize the augmented graphs so they are target-related:

$$\mathcal{L}(G') = \mathcal{I}_{\text{bound}}\left(G'; G\right) - \log p\left(y|f(G')\right). \tag{7}$$

Framework details: After the diffusion model learns the data distribution from the source domain, given a labeled graph G from a specific target domain, DCT perturbs it for D ($D \ll T$) steps. The perturbed noisy graph, denoted by $\tilde{G}^{(D)}$, stays inside the target-specific graph and label space, rather than the noise distribution (at step T). To reverse the noise in it and generate a target-specific augmented example G', DCT integrates the loss function in Eq. (7) into the score function $\mathbf{s}(\cdot, t)$ for minimal sufficient knowledge transfer:

$$\mathrm{d}\tilde{G}^{(t)} = \left[\mathbf{f}(\tilde{G}^{(t)}, t) - g(t)^2 \left(\mathbf{s}(\tilde{G}^{(t)}, t) - \alpha \nabla_{\tilde{G}^{(t)}} \mathcal{L}(\tilde{G}^{(t)})\right)\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{8}$$

where α is a scalar for score alignment between $\mathbf{s}$ and $\nabla\mathcal{L}$ to avoid the dominance of any of them:

$$\alpha = \frac{\|\mathbf{s}(\tilde{G}^{(t)}, t)\|_2}{\|\nabla_{\tilde{G}^{(t)}}\mathcal{L}(\tilde{G}^{(t)})\|_2}. \tag{9}$$

Because $\tilde{G}^{(t)}$ is an intermediate state in the reverse process, the noise in it may fail the optimizations. So, we design a new sampling method named *double-loop sampling* for accurate loss calculation. It has an inner-loop sampling using Eq. (4) to sample $\hat{G}_{(t)}$, as the denoised version of $\tilde{G}^{(t)}$ at the reverse time t. Then $\nabla_{\hat{G}}\mathcal{L}(\hat{G}_{(t)})$ is calculated as an alternative for $\nabla_{\tilde{G}^{(t)}}\mathcal{L}(\tilde{G}^{(t)})$. Finally, an outer-loop sampling takes one step to guide denoising using Eq. (8).

In Figure 1, DCT iteratively creates the augmented graphs (G', y'), updates the training dataset $\{(G_i, y_i)\}$, and trains the polymer property predictor f in the target domain. In each iteration, n graphs of the lowest property prediction loss are selected to create the augmented graphs. The predictor is better fitted to these graphs for more accurate sufficiency estimation of the augmentation.

4 Results and Conclusion

Target Task and metric: The use of polymers in material discovery, such as designing new membranes, has the potential to reduce global energy consumption, greenhouse gas emissions, and pollution Sholl & Lively (2016). We focus on four polymer regression tasks: GlassTemp, MeltingTemp, ThermCond, and O_2Perm. The dataset statistics are presented in Table 1, indicating that the datasets are small and require extra knowledge to train the GNN predictor. They are used to predict different polymer properties such as *glass transition temperature* (°C), *melting temperature* (°C), *thermal conductivity* (W/mK) and *oxygen permeability* (Barrer). GlassTemp and MeltingTemp are collected from PolyInfo, which is the largest web-based polymer database Otsuka et al. (2011). The ThermCond dataset is from molecular dynamics simulation and is an extension of the dataset used in Ma et al. (2022). The O_2Perm dataset is created from the Membrane Society of Australasia portal, consisting of a variety of gas permeability data Thornton et al. (2012). Since a polymer is built from repeated units, researchers often use a single unit graph with polymerization points as polymer graphs to predict properties. Different from molecular graphs, two polymerization points are two special nodes (see "$*$" in Figure 1), indicating the polymerization of monomers Cormack & Elorza (2004). We use the mean absolute error (MAE) to evaluate the model performance. For all the polymer tasks, we randomly split by 60%/10%/30% for training, validation, and test.

Baseline and implementation: Besides GIN, there are three lines of baseline methods: (1) self-supervised learing methods include EdgePred, AttrMask, ContextPred in Hu et al. (2019), InfoMax Velickovic et al. (2019), JOAO You et al. (2021), GraphLoG Xu et al. (2021), and D-SLA Kim et al. (2022), (2) semi-supervised learning methods include InfoGraph Sun et al. (2020) and self-training with selected unlabeled molecular graphs (ST-Real) and generated molecular graphs (ST-Gen), (3) graph data augmentation (GDA) methods include FLAG Kong et al. (2022) and GREA Liu et al. (2022). Implementation details for baselines are in appendix C.1. For DCT, we use source molecules from 113K QM9 Ramakrishnan et al. (2014) and tune three hyper-parameters: the number of perturbation steps $D \in [1, 10]$, the number of negative samples $M \in [1, 10]$, and top-n % labeled polymer graphs of lowest property prediction loss selected for data augmentation.

Preliminary Results and Conclusion: We report the mean and standard deviation of the model performance over 10 runs with randomly initialized parameters, as shown in Table 2. DCT is the best solution and reduces MAE relatively by 1.9% $\sim$ 10.2% compared to the best baseline on different target polymer tasks. In contrast, the self-supervised learning approaches often struggle to transfer knowledge from source molecules to target polymers and underperform the GIN trained only in the target domain with limited polymers. These preliminary results show that using DCT for knowledge transfer across domains is a promising direction.

Acknowledgments

This work was supported in part by NSF IIS-2142827, IIS-2146761, and ONR N00014-22-1-2507.

Table 1: Statistics of datasets for polymer property prediction.

Dataset	# Graphs	Task Type	# Task	Avg./Max # Nodes	Avg./Max # Edges
GlassTemp	7,174	Regression	1	36.7 / 166	79.3 / 362
MeltingTemp	3,651	Regression	1	26.9 / 102	55.4 / 212
ThermCond	759	Regression	1	21.3 / 71	42.3 / 162
O_2Perm	595	Regression	1	37.3 / 103	82.1 / 234

Table 2: Mean(Std) results on tasks of polymer property prediction. The best mean is **bonded**. The best baseline is underlined. The MAE for ThermCond is scaled $\times$ 100.

	# Training Graphs	GlassTemp 4,303	MeltingTemp 2,189	ThermCond 455	O_2Perm 356
	GIN	26.4(0.2)	40.9(2.2)	3.25(0.19)	201.3(45.0)
Self-Supervised	EDGEPRED	27.6(1.4)	47.4(2.8)	3.69(0.50)	207.3(41.7)
	ATTRMASK	27.7(0.8)	45.8(2.6)	3.17(0.32)	179.9(30.8)
	CONTEXTPRED	27.6(0.3)	46.7(1.9)	3.15(0.24)	191.2(35.2)
	INFOMAX	27.5(0.8)	46.5(2.8)	3.31(0.25)	231.0(52.6)
	JOAO	27.5(0.2)	46.0(0.2)	3.55(0.26)	207.7(43.7)
	GRAPHLOG	29.5(1.3)	50.3(3.3)	3.01(0.17)	229.7(48.3)
	D-SLA	27.5(1.0)	51.7(2.5)	2.71(0.08)	257.8(30.2)
Semi-SL	INFOGRAPH	30.8(1.2)	51.2(5.1)	2.75(0.15)	207.2(21.8)
	ST-REAL	26.6(0.3)	42.3(1.2)	2.64(0.07)	256.0(17.5)
	ST-GEN	26.8(0.3)	42.0(0.9)	2.70(0.03)	262.2(10.1)
GDA	FLAG	26.6(1.3)	44.2(2.0)	3.05(0.10)	177.7(60.7)
	GREA	26.7(1.0)	41.1(0.8)	3.23(0.18)	194.0(45.5)
	DCT (Ours)	**23.7**(0.2)	**38.0**(0.8)	**2.59**(0.11)	**165.6**(24.3)

REFERENCES

Peter AG Cormack and Amaia Zurutuza Elorza. Molecularly imprinted polymers: synthesis and characterisation. *Journal of chromatography B*, 804(1):173–182, 2004.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Rachid Hsissou, Rajaa Seghiri, Zakaria Benzekri, Miloudi Hilali, Mohamed Rafik, and Ahmed Elharfi. Polymer composite materials: A comprehensive review. *Composite structures*, 262: 113640, 2021.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, volume 162, pp. 10362–10383. PMLR, 2022.

Dongki Kim, Jinheon Baek, and Sung Ju Hwang. Graph self-supervised learning with accurate discrepancy learning. *Advances in Neural Information Processing Systems*, 2022.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 60–69, 2022.

Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. *Proceedings of the 28th ACM SIGKDD international conference on knowledge discovery & data mining*, 2022.

Ruimin Ma, Hanfeng Zhang, Jiaxin Xu, Luning Sun, Yoshihiro Hayashi, Ryo Yoshida, Junichiro Shiomi, Jian-xun Wang, and Tengfei Luo. Machine learning-assisted exploration of thermally conductive polymers based on high-throughput molecular dynamics simulations. *Materials Today Physics*, 28:100850, 2022.

Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1):1–18, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. Polyinfo: Polymer database for polymeric materials design. In *2011 International Conference on Emerging Intelligent Data and Web Technologies*, pp. 22–29. IEEE, 2011.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

David S Sholl and Ryan P Lively. Seven chemical separations to change the world. *Nature*, 532 (7600):435–437, 2016.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *International Conference on Learning Representations*, 2020.

A Thornton, L Robeson, B Freeman, and D Uhlmann. Polymer gas separation membrane database, 2012.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations*, 2019.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pp. 11548–11558. PMLR, 2021.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pp. 12121–12132. PMLR, 2021.

Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021.

A Appendix

B Additional Method Details

B.1 Upper bounding the mutual information

In Eq. (6), we use a leave-one-out variant of InfoNCE ($\mathcal{I}_{\text{bound}}$) to derive the upper bound of mutual information. We summarize the derivation Poole et al. (2019) here.

$$\begin{aligned}
\mathcal{I}_1(G';G) &= \mathbb{E}_{p(G,G')}\left[\log \frac{p(G'|G)}{p(G')}\right] \\
&= \mathbb{E}_{p(G,G')}\left[\log \frac{p(G'|G)q(G')}{q(G')p(G')}\right] \\
&= \mathbb{E}_{p(G,G')}\left[\log \frac{p(G'|G)}{q(G')}\right] - \mathrm{KL}(p(G')||q(G')) \\
&\leq \mathbb{E}_{p(G,G')}\left[\log \frac{p(G'|G)}{q(G')}\right]
\end{aligned} \tag{10}$$

The intractable upper bound is minimized when the variational approximation $q(G')$ matches the true marginal $p(G')$ Poole et al. (2019). For each G_i, its augmented output G'_i, and $M-1$ negative examples with different labels, we could approximate $q(G'_i) = \frac{1}{M-1}\sum_{j\neq i} p(G'_i|G_j)$. So, we have

$$\begin{aligned}
\mathcal{I}_1(G'_i,G_i) &\leq \log \frac{p(G'_i|G_i)}{\frac{1}{K-1}\sum_{j=1,j\neq i}^{M} p(G'_i|G_j)} \\
&= \log \frac{p(G'_i|G_i)}{\sum_{j=1,j\neq i}^{M} p(G'_i|G_j)} + \log(M-1) \\
&= \mathcal{I}_{\text{bound}}(G'_i;G_i) + \text{constant}
\end{aligned} \tag{11}$$

B.2 Extraction of Statistical Features on Graphs

For each polymer graph, we concatenate the following vectors or values for feature extraction.

- the sum of the degree in the graph;
- the vector indicating the distribution of atom types;
- the vector containing the maximum, minimum and mean values of atoms weights in a molecule or polymer;
- the vector containing the maximum, minimum, and mean values of bond valence.

B.3 Technical Details for Graph Data Augmentation with Diffusion Model

Instantiations of SDE on Graphs According to Song et al. (2021), we use the Variance Exploding (VE) SDE for the diffusion process. Given the minimal noise $\sigma_{\min}$ and the maximal noise $\sigma_{\max}$, the VE SDE is:

$$\mathrm{d}G = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}\mathrm{d}\mathbf{w}, \quad t\in(0,1] \tag{12}$$

The perturbation kernel is derived Song et al. (2021) as:

$$p_{0t}(G^{(t)} \mid G^{(0)}) = \mathcal{N}\left(G^{(t)}; G^{(0)}, \sigma_{\min}^2\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}\mathbf{I}\right), \quad t\in(0,1] \tag{13}$$

On graphs, we follow Jo et al. (2022) to separate the perturbation of adjacency matrix and node features:

$$p_{0t}(G^{(t)} \mid G^{(0)}) = p_{0t}(\mathbf{A}^{(t)} \mid \mathbf{A}^{(0)})p_{0t}(\mathbf{X}^{(t)} \mid \mathbf{X}^{(0)}). \tag{14}$$

The Sampling Algorithm in the Reverse Process for Graph Data Augmentation We adapt the Predictor-Corrector (PC) samplers for the graph data augmentation in the reverse process. The algorithm is shown in Algorithm 1.

Algorithm 1 Diffusion-Based Graph Data Augmentation with PC Sampling

Input: Graph G with node feature $\mathbf{X}$ and adjacency matrix $\mathbf{A}$, the denoising function for node feature $\mathbf{s}_{\mathbf{X}}$ and adjacency matrix $\mathbf{s}_{\mathbf{A}}$, the fine-tune loss $\mathcal{L}_{\textbf{aug}}$, Lagevin MCMC step size β, scaling coefficient ϵ_1
$\mathbf{A}^{(D)} \leftarrow \mathbf{A} + \mathbf{z}_A; \quad \mathbf{z}_A \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{X}^{(D)} \leftarrow \mathbf{X} + \mathbf{z}_X; \quad \mathbf{z}_X \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = D - 1$ **to** 0 **do**
  $\hat{G}_{(t+1)} \sim p_{0t+1}(\hat{G}_{(t+1)}|G^{(t+1)})$ {inner-loop sampling with another PC sampler}
  $\mathbf{S}_A = \frac{1}{2}\mathbf{s}_{\mathbf{A}}(G^{(t+1)}, t+1) - \frac{1}{2}\alpha\nabla_{\mathbf{A}^{(t)}}\mathcal{L}_{\textbf{aug}}(\hat{G}_{(t+1)})$
  $\mathbf{S}_X = \frac{1}{2}\mathbf{s}_{\mathbf{X}}(G^{(t+1)}, t+1) - \frac{1}{2}\alpha\nabla_{\mathbf{X}^{(t)}}\mathcal{L}_{\textbf{aug}}(\hat{G}_{(t+1)})$
  $\tilde{\mathbf{A}}^{(t)} \leftarrow \mathbf{A}^{(t+1)} + g(t)^2\mathbf{S}_A + g(t)\mathbf{z}_A; \quad \mathbf{z}_A \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Predictor for adjacency matrix}
  $\tilde{\mathbf{X}}^{(t)} \leftarrow \mathbf{X}^{(t+1)} + g(t)^2\mathbf{S}_X + g(t)\mathbf{z}_X; \quad \mathbf{z}_X \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Predictor for node features}
  $\mathbf{A}^{(t)} \leftarrow \tilde{\mathbf{A}}^{(t)} + \frac{\beta}{2}\mathbf{S}_A + \epsilon_1\sqrt{\beta}\mathbf{z}_A; \quad \mathbf{z}_A \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Corrector for adjacency matrix}
  $\mathbf{X}^{(t)} \leftarrow \tilde{\mathbf{X}}^{(t)} + \frac{\beta}{2}\mathbf{S}_X + \epsilon_1\sqrt{\beta}\mathbf{z}_X; \quad \mathbf{z}_X \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Corrector for node features}
end for
return $G' = (\mathbf{A}^{(0)}, \mathbf{X}^{(0)})$

C Additional Experiment Details

C.1 Baselines and Implementation

When implementing GIN Xu et al. (2019), we tune its hyper-parameters for different tasks with an early stop on the validation set. We generally implement pre-training baselines following their own setting. The pre-trained GIN models with self-supervised tasks such as EdgePred, AttrMask, ContextPred in Hu et al. (2019), InfoMax Velickovic et al. (2019) are available. So we directly use them. For other self-supervised methods, we implement their codes with default hyper-parameters. Following their settings, we use 2M ZINC15 Sterling & Irwin (2015) to pre-train GIN models for polymer property prediction. For self-training with real unlabeled molecules and InfoGraph Sun et al. (2020), we use 113K QM9 Ramakrishnan et al. (2014). For self-training with generated molecules graphs, we train the diffusion model Jo et al. (2022) on the real QM9 dataset and then produce the same number of generated unlabeled molecules. To train the diffusion model in our DCT, we also use QM9 Ramakrishnan et al. (2014).