Hierarchical-Attention Graph Learning for Molecular Solubility Prediction

Yangxin Fan
Case Western Reserve University
Cleveland, Ohio, USA
yxf451@case.edu

Yinghui Wu
Case Western Reserve University
Cleveland, Ohio, USA
yxw1650@case.edu

Roger French
Case Western Reserve University
Cleveland, Ohio, USA
rxf131@case.edu

Danny Perez
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
danny_perez@lanl.gov

Michael Taylor
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
mgt16@lanl.gov

Ping Yang
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
pyang@lanl.gov

Abstract

Molecular solubility (or simply solubility) quantifies the concentration of a molecule that can dissolve in a given solvent. Accurate prediction of solubility is essential for optimizing drug efficacy, improving chemical and separation processes, controlling pollution and waste management, among many other industrial and research applications. However, predicting solubility from first principles remains a complex and computationally intensive physicochemical challenge. Recent successes of graph neural networks in molecular graph learning tasks inspire us to propose HASolGNN, a novel hierarchical-attention graph neural networks for accurate prediction of solubility. (1) HASolGNN adopts a novel three-level hierarchical attention framework to characterize atom level, molecular level, and interaction-graph level features. This hierarchical design not only comprehensively characterizes intricate structural details of individual molecules but also effectively captures the complex intermolecular interactions involved in solute dissolution within solvents. HASolGNN encodes solute and solvent molecular graphs separately with atom embedding block and molecular embedding block. (2) To best harvest sparse yet diversified annotated datasets ("small data"), we further introduce HASolGNN-LLMs, a variant of HASolGNN enhanced by the Large Language Model (LLM), where it plays a role as an annotator and a feature enricher for feature alignment. Using three curated solubility benchmark datasets, our experiments verified that HASolGNN outperforms the state-of-the-art methods including AttentiveFP and MFGNN in solubility prediction. Moreover, we verify that HASolGNN-LLMs significantly improves solubility prediction in "small data" scenarios.

1 Introduction

The molecular solubility of a solute in a solvent, also known as molecular-level mixability via intermolecular interaction, is broadly relevant to many areas of chemistry, physics, geochemistry, biochemistry, and pharmaceutical science. It is also of great importance in a variety of applications, including nuclear waste separation [14, 67], environmental pollutions control [37], development of advanced materials in the semi-conductor industry [9, 34], autonomous robotics synthesis [54, 56], crystallization [33], and protein ligand bonding in the biomedical field [16, 48]. Particularly, aqueous solubility refers to the solubility of a solute in water. It plays an essential role in pharmaceutical science [7, 13, 24] since (1) accurate prediction of solubility is critical for selecting promising drug candidates during the screening process, and (2) all drugs in the body exert their therapeutic effects in the form of aqueous solutions which means lower solubility diminishes both their efficacy and bioavailability. Therefore, high-precision computational methods for solubility prediction can substantially decrease the experimental costs and time associated with drug development [35] while enabling chemists to develop formulations that maximize drug efficacy and improve patient outcomes.

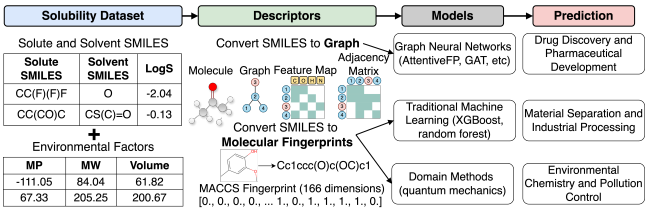


Figure 1: Solubility Prediction Pipeline with Four Modular Components (solubility prediction models are categorized into three major types, with critical applications spanning fields such as material separation and drug discovery).

Example 1.1. Fig. 1 shows a a pipeline of solubility prediction task, consisting of four components: solubility dataset, descriptors, models, and solubility prediction. Solubility data typically refer to the SMILES representation [29] of solute and solvent pairs, the values of LogS (log-scale solubility) measured under certain environmental factors, as well as the environmental factors such as Melting Point (MP), Molecular Weight (MW), and Volume. The pipeline next converts the SMILES to (1) molecular graphs and (2) molecular fingerprints (physiochemical features). Molecular graphs will be fitted to Graph Neural Networks-based methods (*e.g.,* [2, 10, 30, 65]), while molecular fingerprints will be fitted to traditional machine learning and domain methods (*e.g.,* [25, 26, 40, 45]). Next, the models predict solubility for critical applications in areas such as drug discovery and material separation.

State-of-the-Art. Solubility prediction has been a long-standing challenge for chemical and data science community. A host of methods have been developed to predict the solubility of molecular systems. Recent approaches fall into three categories: (1) *Domain-specific methods*, rooted in principles of quantum mechanics and thermodynamics; (2) Traditional *ML-based methods*, which leverage established regression models and ensemble learning to predict

solubility, and more recently, (3) *Graph neural networks (GNN)-based* methods, which utilize molecular graphs to model atoms and bonds and train GNNs as regression or classifiers.

(1) *Domain-specific* methods based on quantum mechanics and thermodynamics primarily adhere to the Quantitative Structure-Property Relationship (QSPR) framework [36]. They regress solubility against a carefully selected set of molecular descriptors, capturing the structural and physicochemical properties of compounds. Depending on the underlying theoretical foundation, such methods employ a range of mathematical models, including differential and partial differential equations, to predict solubility [25, 40, 50].

(2) Among *ML-based methods*, boosting methods XGBoost [26] and random forest [45] have been widely applied. Other ML models include multi-layer perceptron (MLP) [4] and artificial neural networks (ANNs) [57]. These methods take molecular fingerprints as the input features, leveraging the encoded molecular-level structural information to predict solubility.

(3) Several GNNs-based methods were developed, which exploit graph convolutional networks (GCNs) [10, 30], gated graph neural networks (GGNNs) [3, 46], node-level attention-based graph attention networks (GATs) [31, 70], and molecular representation learning integrating both node- and graph-level attention mechanisms (AttentiveFP) [2, 65]. Among these methods, AttentiveFP has demonstrated superior performance, establishing itself as the current leading approach in GNNs-based solubility prediction [2].

Challenges & Opportunities. While aforementioned approaches have been applied for solubility prediction, it remains challenging due to several data challenges, which also indicate opportunities.

(1) *Multi-level features*: Most methods fit models on molecular fingerprints alone. The sequential encoding often overlooks high-value features at atom and bond ("edges" connecting two atoms) level, and topological features encoding their interactions within the molecular. On the other hand, such features may be scattered in small, heterogeneous datasets, among which few are annotated or labeled (a case of "small data").

(2) *Inter-molecular interaction*: Solubility is determined by a dynamic interaction process between solute and solvent pairs. Existing GNN-based methods often compromise to model the solubility by jointly learning atom and bond-level representation of solute alone, but lack the expressiveness to capture the interactions between general cases with possibly unseen solute-solvent pairs.

(3) Existing domain-specific methods are often constrained by domain hypothesis and models, mostly lead to case-by-case analysis. Most methods are specialized for aqueous solubility, or limited to fixed groups of solvents. Hence existing methods are often hard to be generalized for solubility prediction.

In response, we advocate to develop a solubility predictive framework as a "general recipe" to be broadly applied across diverse solute-solvent systems. Such a framework should be able to (a) best harvest a "hierarchy" of features from atom-, bond-, molecular-, as well as environmental, among other external features; (b) characterize inter-molecular interaction for more comprehensive and accurate modeling of solubility, and (c) easily integrate, annotate and align heterogeneous features with few or no annotated data.

Contribution. To address aforementioned challenges, we propose HASolGNN, a hierarchical-attention graph neural networks for solubility prediction. HASolGNN combines the strengths of physics-informed machine learning and graph neural networks method, that can jointly exploit both the rich topological information from the atoms and bonds in molecular graphs, and the molecular interactions between the solute and solvent.

(1) We propose **HASolGNN**, a novel framework that effectively captures hierarchical, multi-levels interactions and patterns among atom, bond, and (inter- and intra-) molecular. HASolGNN enables solubility prediction across a wide range of solute-solvent pairs, irrespective of solvent types. HASolGNN achieves above through the integration of hierarchical attention mechanisms across three key components: the atom embedding (AE) block, the molecular embedding (ME) block, and the interaction-graph embedding (IE) block. Experimental results demonstrate that HASolGNN significantly outperforms the state-of-the-art graph neural networks methods, establishing new benchmarks in performance.

(2) We further propose **HASolGNN-LLMs** that integrates large-language models (LLMs) as a modular component, leveraging contrastive learning for fusion with HASolGNN to address the "small dataset" challenges frequently encountered by the scientific community. Our experiments show that HASolGNN-LLMs yield substantial improvements in solubility prediction under such data-scarce conditions, offering a potential solution to this common limitation.

(3) We conduct a comprehensive evaluation of the performance of diverse graph neural network variants on three extensive and representative public solubility datasets, benchmarking our method's performance against the state-of-the-art methods. Our efforts offer valuable insights into the use of GNNs for solubility prediction that benefit both the chemistry and computer science communities.

Related Work. Besides aforementioned methods, we summarize other related work below.

Graph Learning for System-level Regression. Graph Neural Networks (GNNs)[63] have been extensively studied with promising performance for a variety of applications. A GNN model may exploit spectral-based, attention-based, and spatial-based convolutions to manipulate node features, edge features, and graph-level representations. Through iterative message propagation, a k-layers GNNs updates node and edge representations by aggregating information from neighboring nodes up to k-hops away. Notable GNN variants include Graph Convolutional Network (GCN) [28], GraphSAGE [20], Graph Attention Network (GAT) [59], and Graph Isomorphic Network (GIN) [66].

While GNNs are widely adopted for graph analytics such as node classification or link prediction, the use of GNNs for system modeling with multiple participating molecules has been comparatively less explored. On example of system-level regression is molecular solubility prediction, which requires effective modeling of interactions between two molecular graphs as a dynamic system. As aforementioned, existing GNNs-based methods, while being adaptable for graph-based molecular property prediction, rely on graph-level pooling strategies to produce numeric predictions for solubility of a solute-solvent system. Therefore, these approaches overlook the potentially complex interactions between molecules,

as they remain constrained to explicitly model the interactions among different molecules in a dynamic system.

Graph Learning-Based Molecular Property Prediction. Graph learning have recently gained significant attention for molecular property prediction. Recurrent graph neural networks (RGNNs) [19, 38, 42, 52, 61] were among the first GNNs utilized for molecular property prediction. RGNNs iteratively apply shared weight matrices, enabling the model to capture dependencies over multiple iterations. Conv-GNNs [12, 17, 21, 51, 60] introduce iteration-specific weights, enhancing flexibility and expressiveness. Specifically, spectral Conv-GNNs, operating in spectral domain using graph Laplacian transformations, and spatial Conv-GNNs, directly aggregating features from neighboring nodes. Beyond RGNNs and Conv-GNNs, several architectural innovations include advance pooling strategies [11, 68] and skip-connections [39, 41], which can be integrated into any type of GNNs to further improve feature aggregation. Additionally, architecturally distinct GNNs [55, 71] can also be used to integrate novel mechanisms for molecular system modeling.

2 Preliminary

Molecule as a Graph. We represent a molecule as a graph $G = (V, E, X, Y, \mathcal{F})$, which specifies the following:

- V denotes the set of nodes, with each node representing an atom in the molecule;
- E refers to the set of edges, corresponding to the bonds between atoms;
- X represent the node features including atomic number, degree (number of bonds connected to the atom), formal charge, number of unpaired electrons (radical electrons), hybridization state (e.g., sp^3, sp^2, sp), aromaticity, number of implicit hydrogen atoms, chirality and chirality type [65]; and x_u represent the node feature vector of node $v \in V$;
- Y represent the bond features including four bond types (single, double, triple or aromatic), conjugation (binary, 1 if the bond is conjugated, 0 otherwise), ring (1 if the bond is part of a ring, 0 otherwise), and stereo (a one-hot encoded vector of length 4 to represent the stereochemistry of the bond) [62]; y_{vu} represent the bond feature vector of the bond connecting node v with node u; and
- $\mathcal{F}$ denotes the graph-level features, encompassing molecular fingerprints capturing the molecule's structural and functional characteristics (e.g. molecular weight, total charge), along with environmental factors such as temperature and pH, which influence the molecule's behavior.

This representation provides a comprehensive depiction of the molecule, capturing local atomic and bond-level details while integrating global and environmental context, enabling robust modeling of molecular properties and interactions.

System-level Regression. We define system-level regression as the task of predicting a property arising from interactions among p molecules. These properties arise from complicated interactions among multiple molecules in a system, such as those in the solute-solvent systems. The input consists of a series of molecular graphs $G_1, G_2, \ldots, G_p$ and the regression model applies pooling strategies and fusion or concatenation of graph embeddings to capture the system-level molecular interactions. The goal is to predict a numeric value for a system-level property, such as solubility, affinity, toxicity, or side effect, that should depend on these molecular interactions.

We formulate solubility prediction as a special case of system-level regression task with $p = 2$, representing the two interacting molecules involved: the solute (G_1) and the solvent (G_2). Our goal is to train a GNN model M, which takes as input G_1 and G_2 to minimize solubility prediction errors $\mathcal{L}_S(G_1, G_2, L, \theta)$ where L and θ denote ground-truths and the set of parameters in M respectively.

M differs from prior GNN-based approaches in the following: (1) M jointly learns representations at different levels in G_1 and G_2. By employing atom- and molecule-level attentions and treating the molecule-level representation as a supernode connected to all atoms within the molecule, it can capture complex atomic and bond interactions while simultaneously modeling the "non-local" interactions within the molecule. (2) M aggregates graph-level fingerprint features derived from G_1 and G_2 and integrate them as node features, all within an interaction graph block (to be discussed). This representation enables M to explicitly model the molecular interactions between the solute and the solvent.

3 HASolGNN Framework

We next introduce the HASolGNN architecture and its components.

Model Architecture. We start with the architecture of HASolGNN, as illustrated in Fig. 2. It consists of the following key modules.

(1) **Featurization and Input Layer**: HASolGNN converts the SMILES representations of solute S_1 and solvent S_2 into the featurized solute graph $G_1 = (V_1, E_1, X_1, Y_1, \mathcal{F}_1)$ and solvent graph $G_2 = (V_2, E_2, X_2, Y_2, \mathcal{F}_2)$. The input layer standardizes them into $\hat{G_1}$ and $\hat{G_2}$ by combining their corresponding node and bond features;

(2) **Molecular Fingerprint Generation Module (MFGM)**: HASolGNN fits $\hat{G_1}$ and $\hat{G_2}$ into MFGM. Each MFGM is composed of one Atom Embedding (AE) block and two Molecule Embedding (ME) blocks. The AE block processes G_1 or G_2 while the ME blocks takes as input G_1^M or G_2^M. G_1^M and G_2^M are augmented graphs each containing a supernode, connected to all the atoms in G_1 or G_2. The embeddings produced by the AE block are fed into the first ME block. In contrast, the second ME block directly utilizes the initial node and bond features of the molecular graphs;

(3) **Embeddings Fusion**: HASolGNN leverages a novel fusion mechanism to combine the embeddings representing molecule level representation of the solute and solvent from the two MFGMs with graph-level features $\mathcal{F}_1$ and $\mathcal{F}_2$ to generate the molecular fingerprints MF_{solute} for solute and $MF_{solvent}$ for solvent;

(4) **Interaction Graph**: HASolGNN constructs the Interaction Graph comprising two nodes: G_1, representing solute with MF_{solute} as its node features, and G_2, representing the solvent with $MF_{solvent}$ as its node features. The Integration-graph Embedding (IE) block processes the Interaction Graph to produce the system-level fingerprint MF_{IG} which is passed to the output layer to predict solubility.

Justification of design. It is essential for HASolGNN to achieve a smooth and hierarchical encoding from the molecular graphs to the system-level representation for solubility. By contrast, traditional GNNs methods address this as a graph-level regression problem,

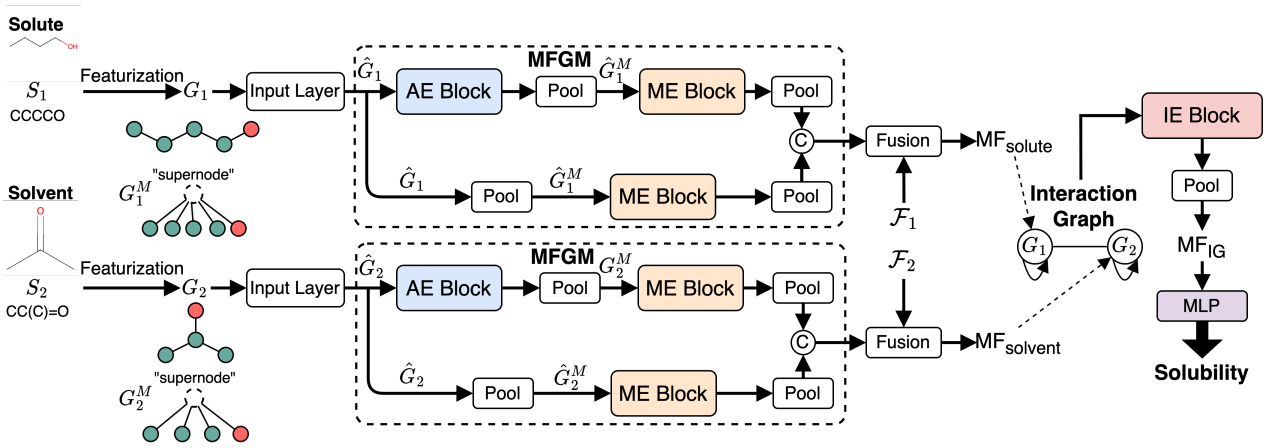


Figure 2: Proposed HASolGNN Framework (MFGM: molecular fingerprint generation module, molecular graphs G_1 of solute and G_2 of solvent fitted into different MFGM each consisting of one AE Block and two ME Blocks, IE Block fits on the interaction graph and outputs the system-level representation MF_{IG} fitted into the output layer to generate solubility prediction).

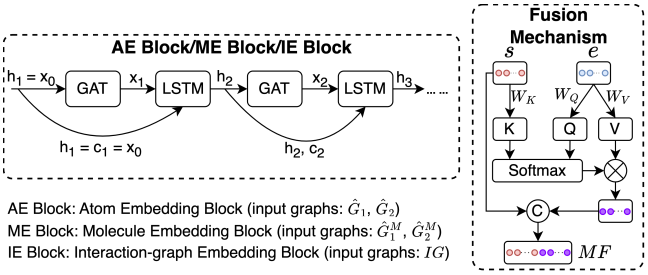


Figure 3: Details of Proposed Atom Embedding (AE) Block, Molecule Embedding (ME) Block, and Interaction-graph Embedding (IE) Block (a series of GAT and LSTM Layers) and Proposed Fusion Mechanism with Cross Attention.

where solubility prediction is generated by applying a pooling operation to the concatenated node features of solute and solvent graphs, extracted from the final layer of the GNNs. However, this often leads to significant information loss due to the high information compression in capturing intricate interactions between solute and solvent, ultimately degrading performance. This limitation is experimentally validated in Section. 5. Our proposed hierarchical embedding architecture ensures a more seamless encoding process, offering a far more effective way to model the solubility prediction by capturing atomic, molecular, and system-level information.

We next detail the key components of HASolGNN.

Input Layer. For each node $v \in V$, the input layer concatenates the node features from the neighboring nodes and edge features from the incident edges and unifies the node representations across all nodes by taking both initial atom and bond features into account. The input layer standardizes the input and ensures that both the initial bond and atom features participate in the following message passing. The input layer generates the updated molecular graphs, incorporating enhanced node features derived from the original node and edge attributes. These updated graphs $\hat{G}_1$ and $\hat{G}_2$ are then passed to the downstream AE Block and one of the ME Blocks in the MFGM. Formally, we represent the input layer as follows:

$$
\begin{aligned}
h_v^0 &= \mathrm{ReLU}(W_{fc1} x_v), \ \forall v \in V; \\
h_u^0 &= \mathrm{ReLU}\big(W_{fc2}\mathrm{CONCAT}(x_u, y_{vu})\big), \ \forall u \in N(v); \\
a_{vu}^0 &= \mathrm{Softmax}\big(\mathrm{LeakyReLU}(W[h_v^0, h_u^0])\big); \\
h_v^1 &= \mathrm{GRU}\Big(\mathrm{ELU}\big(\textstyle\sum_{u \in N(v)} a_{vu} W h_u^0\big), h_v^0\Big)
\end{aligned}
\tag{1}
$$

where W_{fc1}, W_{fc2}, and W are learnable weight matrices and ELU denotes the Exponential Linear Unit activation function.

Atom Embedding (AE) Block. As illustrated in Fig. 3, the Atom Embedding (AE) Block consists of an iterative process comprising (1) a message-passing phase, implemented using a Graph Attention Network (GAT) layer, followed by, (2) a readout phase utilizing a Long Short-Term Memory (LSTM) layer that performs information filtering and models long-range dependencies. AE Block fits directly on the output of input layer $x_0 = h^1$, the updated node embedding derived by the input layer. At the first iteration, both the hidden state h_1 and cell state c_1 are initialized to x_0. At the t-th iteration ($t > 1$), h_t is passed to the t-th GAT layer to compute the x_t, which serves as the input to the t-th LSTM layer. The t-th LSTM layer updates its hidden state to h_{t+1} and cell state to c_{t+1}. We represent the iterative refinement process as follows:

$$
(h_{t+1}, c_{t+1}) = \mathrm{LSTM}\Big(\mathrm{GAT}\big(h_t, A_G\big), \big(h_t, c_t\big)\Big), t \in [1, k] \tag{2}
$$

After the k-th iterations, the final hidden state h_{k+1} will be forwarded to the downstream tasks.

Molecular Embedding (ME) Block. With in each MFGM, two ME blocks that process synthetic graphs with identical topology, where a single supernode connects to all atoms in the molecular graph. However, the node features differ between these blocks. For the top ME Block, node features are computed from the output of the AE Block such that 1) the initial supernode embedding is obtained by pooling h_{k+1} and 2) the node features of all atoms are directly inherited from the their embeddings in h_{k+1}. In contrast, the bottom ME Block constructs node features from the output of the input layer such that 1) the initial supernode embedding is derived by

pooling of h^1 and 2) the node features of all atoms directly inherited from the updated molecular graphs from the input layers.

Fusion Mechanism. We propose a novel fusion mechanism to integrate graph-level features $\mathcal{F}_1$ and $\mathcal{F}_2$ that capture contextual information such as environmental factors (EFs) with the embeddings from the output of MFGM that conveys structural relationships. Our fusion mechanism, illustrated in Fig. 3, comprises the two steps: 1) cross-attention, and 2) concatenation.

The cross-attention module consists of learnable parameters: query W_Q, key W_K, and value W_V. These are used to compute the cross-attention scores between the structural GNNs embeddings and the EFs. Specifically, it calculates the cross-attention score between the structural embedding vector s and the environmental factors vector e, where s serves as the key, representing the referenced information. The resulting attention weight α transforms e into e' which will be then concatenated with s to derive the molecular fingerpints MF_{solute} for solute and $MF_{solvent}$ for solvent. Formally, we represent fusion mechanism as follows:

$$\alpha = \text{Softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big); \quad e' = \alpha V; \\ MF = \text{CONCAT}\big(s, e'\big) \tag{3}$$

Where $Q = W_Q e$, $K = W_K s$, and $V = W_V e$. The term d_k denotes the dimensionality of the key embeddings s.

Interaction-graph Embedding (IE) Block. HASolGNN constructs the interaction by creating two interconnected nodes, each representing the solute and solvent, respectively. HASolGNN fuses the outputs of the two MFGMs with environmental factors (EFs) to derive the molecular fingerprint for the solute MF_{solute} and the molecular fingerprint of solvent $MF_{solvent}$. These molecular fingerprints are assigned as the node features for their corresponding nodes. The IE Block then processes the integration graph as the computation graph to compute the system-level fingerprint, MF_{IG}.

Hierarchical Attention Mechanism. HASolGNN employs GATs at every iteration of the messaging-passing phase within the AE, ME, and IE blocks, as well as the input layer. This design establishes a three-level hierarchical attention mechanism: 1) node-level and bond-level attentions to capture fine-grained features from the molecular graphs; 2) molecule-level attention derived from synthetic graphs, where each graph includes a supernode connected to all atoms in the corresponding molecular graph; and 3) system-level attention to extract high-level interactions from the interaction graph. HASolGNN leverages this hierarchical messaging passing framework to encode information in a progressively compact manner, transforming detailed atom and bond features progressively into a compact and structured system-level representation.

Loss Function. In the model training, the set of parameters θ of HASolGNN model M is optimized by minimizing the errors between the solubility prediction of HASolGNN and the ground-truth values:

$$\mathcal{L}_S(G_1, G_2, L, \theta) = \frac{1}{N}\sum_{i=1}^{N}(P_{M(G_1,G_2,\theta)}(i) - L(i))^2 \tag{4}$$

where N represents the number of solute-solvent pairs in the training set, P_M denotes the predictions made by HASolGNN, and L corresponds to the ground-truth solubility values.

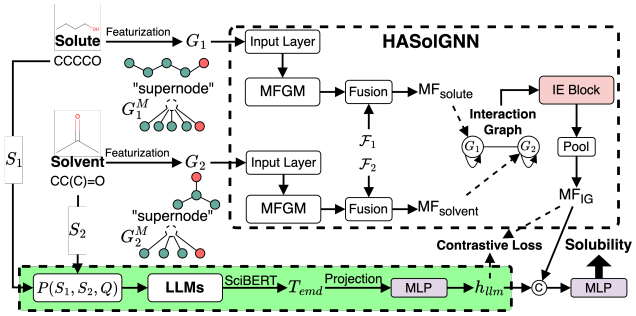


Figure 4: Proposed HASolGNN-LLMs Framework. The pluggable LLMs module, marked in green, aligns the embeddings h_{llm} from the LLMs with the MF_{IG} from HASolGNN with the unsupervised contrastive loss $\mathcal{L}_c$. The prompt $P(S1, S2, Q)$ is used to query LLMs such as GPT-4.

Cost Analysis. The training complexity of HASolGNN is $O(ek|E|dF^2 + ek|V|F^2)$ and the inference complexity is $O(k|E|dF^2 + k|V|F^2)$. Here $|E| = |E_1| + |E_2|$ and $|V| = |V_1| + |V_2|$, and e, k, F, and d refers to the epochs number, number of iterations in AE Block, number of features per node, and the maximum node degree from a node in G_1 or G_2, respectively. As $|V|$, $|E|$, k, F, d are typically small, the training and inference of HASolGNN can be considered in low polynomial time. We present the detailed analysis in the Appendix.

4 LLM-Enhanced HASolGNN

HASolGNN-LLMs leverages LLMs to address the challenge posed by small datasets by playing the following roles: (1) feature enricher: expanding the feature space and enhancing molecular representations to maximize the utility of available data for solubility prediction; and (2) pseudo-annotator: providing qualitative numerical estimations of solubility to supplement small datasets. Building on this, we introduce HASolGNN-LLMs. As shown in Fig. 4, it integrates an LLM module and employs contrastive learning-based fusion to combine embeddings from the LLM with those from HASolGNN.

Architecture Details. We introduce the three primary components of the pluggable LLMs module integrated into HASolGNN-LLMs.

Generating Textual Descriptions. For each solubility prediction between a solute and a solvent, we leverage GPT-4 [1] to generate a textual description of solubility estimations based on their structural and chemical properties of both molecules. Specifically, we incorporate the SMILES strings of the solute S_1 and solvent S_2 to query GPT-4 with the following template $P(S_1, S_2, Q)$[1]:

> "I have two molecules represented by SMILES strings: Solute: [S_1]; Solvent: [S_2]. *(Optional)*: we also know [**Q**]). Based on their **molecular structures**, provide specific, compact, and tailored descriptions about the **solubility** of the solute in the solvent. Discuss any **chemical properties** that influence solubility, such as **polarity**, **hydrogen bonding**, **molecular size**, or others. "

Here Q is optional context for user-specified, task-specific domain knowledge such as facts about the chemical structure [23, 48].

[1]We showcase example prompts and LLM responses in Appendix.

Deriving the Embedding h_{llm}. After the textual descriptions are generated by the LLM, we utilize SciBERT [5] to transform textual descriptions into global-level text embeddings T_{emd}, by extracting the [CLS] [5] token embedding, which provides a compact summary representation of all sentences within the descriptions. Given the high dimensionality of T_{emb} compared to the output of HASolGNN, we employ a multi-layer perceptron (MLP) to project T_{emb} into a low-dimensionality latent space, h_{llm}, aligning its dimensionality with that of the MF_{IG}. This step ensures compatibility between the two embedding spaces. We represent this process as follows:

$$h_{llm} = \text{MLP}\Big(\text{SciBERT}\big(\text{LLMs}\big(P(S_1, S_2, Q)\big)\big)\Big) \tag{5}$$

Embeddings Alignment. To integrate the embeddings from HASolGNN with the LLMs module effectively, we align the system-level HASolGNN embeddings MF_{IG} with the textual embeddings h_{llm} derived from the LLMs module. We employ an unsupervised contrastive loss, L_c (please refer to Eqn. 6), to minimize the distance between MF_{IG} and h_{llm} for the same solute-solvent pair. To further enhance alignment, we incorporate a negative sampling strategy [43] to construct negative solute-solvent pairs (τ, τ'), where $\tau = (S_1, S_2)$ and $\tau' = (S_1', S_2')$ such that $\tau \neq \tau'$. For these negative pairs, L_c maximizes the distance (minimizes the similarity) between MF_{IG} of τ and h_{llm} of τ', ensuring proper separation between positive and negative pairs. This alignment allows HASolGNN-LLMs to differentiate between positive and negative solute-solvent pairs, improving its predictive and representational capabilities.

In scenarios with small datasets, the quality of MG_{IG} embeddings may be limited due to the lack of sufficient training data. By incorporating h_{llm}, HASolGNN-LLMs expands feature space, potentially improving solubility prediction accuracy. This is because LLMs module serves a dual role: as a feature enricher, enhancing the representation space; and as a pseudo-annotator, providing qualitative numerical estimations to supplement small datasets.

Loss Function. The loss function of HASolGNN-LLMs $\mathcal{L}$ comprises two components: (1) a supervised loss $\mathcal{L}_S$ that captures the solubility prediction errors from HASolGNN; and (2) an unsupervised contrastive loss $\mathcal{L}_c$ to ensure the proper alignment between the system-level MF_{IG} from HASolGNN and the projected h_{llm} derived by LLMs module. Mathematically, we formulate $\mathcal{L}$ as follows:

$$\begin{aligned} \mathcal{L}(G_1, G_2, L, \theta, S_1, S_2, Q) &= \mathcal{L}_S(G_1, G_2, L, \theta) + \\ &\lambda \mathcal{L}_c(MF_{IG}(G_1, G_2, \theta), h_{llm}(S_1, S_2, Q)); \\ \mathcal{L}_c &= \frac{1}{2N} \sum_{i=1}^{N} \left[y_i \cdot \mathrm{D}_i^2 + (1 - y_i) \cdot \max(0, m - \mathrm{D}_i)^2 \right] \end{aligned} \tag{6}$$

Here, D_i is the distance equal to one minus the cosine similarity between the i-th MF_{IG} and i-th h_{llm}, m is a margin value, λ denotes the balancing factor between $\mathcal{L}_S$ and $\mathcal{L}_c$, and y_i indicates whether the pair is positive ($y_i = 1$) or negative ($y_i = 0$).

5 Experimental Study

We experimentally evaluate the performance of HASolGNN across three solubility benchmark datasets, comparing it with ten GNNs baseline models in terms of prediction accuracy. Additionally, we verify the effectiveness of HASolGNN-LLMs in overcoming the "small dataset" challenge by expanding the feature space.

5.1 Experimental Setup

Evaluation Metrics. We evaluate the performance of our solubility predictions and baselines using Mean Absolute Error (MAE).

Datasets. To validate the effectiveness of our model, we conduct experiments on three large-scale benchmark solubility datasets.

(1) **Exp-DB** [27]: This dataset contains 11,637 experimentally measured solubility values at temperatures of 298 K (±2 K). Each value corresponds to a unique solute-solvent pair, making it the largest dataset in terms of unique pair counts among the benchmarks. (2) **BigSolDB** [29]: The largest solubility dataset in terms of sample sizes, BigSolDB covers a wide range of compounds in both organic solvents and water. It includes 54,273 individual solubility values for 830 unique molecules and 138 distinct solvents, measured over a temperature range of 243.15 to 403.15 K at atmospheric pressure. Each unique solute-solvent pair has ranging from 1 to 20 different solubility values measured at different temperature. Despite its size, BigSolDB contains fewer unique solute-solvent pairs, with only 4,964 unique pairs. (3) **MolMerger** [49]: It comprises 6,975 unique solute-solvent pairs, each with a measured solubility value at temperatures near 273 K. The MolMerger dataset integrates data from three distinct sources: 4,964 values from BigSolDB [29], 1,093 values from BNNLabs Solubility [8], and 972 values from ESOL [15].

For Exp-DB and MolMerger, where each solubility corresponds to an unique solute-solvent pair, we partition the datasets by solubility values, 60% for training, 20% for validation, and 20% for testing. Since each solubility value corresponds to an unique pair, this split naturally follows the inductive learning setting. For BigSolDB, we perform two types of dataset splits: (1) transductive-learning setting: the split is based on solubility values, allowing the same solute-solvent pair to appear in both training and testing, albeit with different temperature measurements; (2) inductive-learning setting: the split is based on the unique solute-solvent pairs, ensuring that all pairs in the test dataset are completely unseen during the training. In this case, we partition the BigSolDB by unique pairs, 60% solute-solvent pairs for training, 20% for validation, and 20% for testing.

Baselines. We compare HASolGNN with 10 baselines: (1) *GCN* [28]: Graph convolutional networks with convolutional layers and a localized first-order approximation of spectral graph convolutions; (2) *GAT* [59]: Graph Attention Networks with node-level attention mechanism, using self-attention to learn the attention scores for the neighbors; (3) *GraphSAGE* [22]: learns node embeddings by sampling and aggregating features from a node's local neighborhood; (4) *GIN* [66]: Graph Isomorphic Network aggregates node features using a sum function, followed by a MLP to update embeddings. It is provably as powerful as the Weisfeiler Lehman graph isomorphism test. (5) *GatedGNN* [32]: uses gated recurrent units (GRUs) to propagate information across nodes in a graph over multiple time steps, enabling the network to capture complex dependencies; (6) *ResGatedGNN* [58]: extends the tree-LSTM to arbitrary graphs and multiple layers by leveraging the vanilla graph ConvNet architecture and the edge gating mechanism. It incorporates residual networks into multi-layer gated graph ConvNets. (7) *CGCN* [64]: Crystal Graph Convolutional Neural Network (CGCN)

is a graph convolutional neural networks framework to learn graph-level properties from the connection of nodes in the graph, providing a universal and interpretable graph-level representation. (8) *GraphTransformer* [53]: utilizes multi-head attention to enable attentive information propagation between nodes, enhancing graph learning capabilities. (9) *MFGNN* [17]: MFGNN introduces a GNN that allows end-to-end learning of prediction pipelines whose inputs are graphs of arbitrary size and shape. MFGNN generalizes standard molecular feature extraction methods based on circular fingerprints. (10) *AttentiveFP* [2]: the SOTA GNN molecular graph representation learning method. AttentiveFP leverages both atom- and molecule-level attention mechanisms by stacking graph attention networks (GAT) with gated-recurrent units (GRUs) to better capture the hierarchical molecular structures. AttentiveFP is capable of extracting non-local intramolecular interactions that are intractable for most graph-based representations.

We trained all baselines with consistent setting (data splits, hyperparameter tuning, etc.) for fair comparisons. Only AttentiveFP adopted two-level (atom and molecular) representation. For the rest, we adopted as a routine the node-level features (SMILES).

Hyperparameter Tuning. We perform a grid search over the validation loss to find the optimal set of hyperparameters used in HASolGNN, following the methodology outlined in [47]. We varied the number of iterations k of the AE Block in the set $\{1, 2, 3, 4\}$, the number of iterations t of the ME Block in the set $\{1, 2, 3, 4\}$, the number of iterations h of the IE Block in the set $\{1, 2, 3, 4\}$, the number of epochs e in the set $\{25, 50, 100, 150, 200\}$, the learning rate from $\{0.001, 0.01, 0.05, 0.1\}$, the contrastive balancing term λ in the set $\{0.25, 0.5, 1, 5, 10, 25, 50, 100\}$. Within GPT-4, we vary temperature in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and top_p in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Based on the validation loss, we choose k equal to 3, t, h, and λ to be 3, 1, and 5 respectively, epochs to be 100, the learning rate to be 0.01, temperature to be 0.3 and top_p to be 0.7.

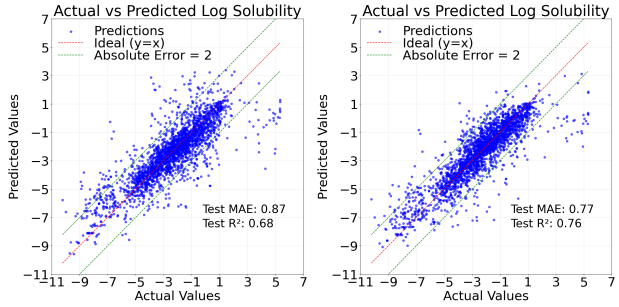
5.2 Experimental Results

First, we analyze solubility prediction errors of HASolGNN compared to all the baselines across all datasets. Next, we study how the incorporating the LLMs Module can improve the solubility prediction over small dataset. Then, we test the robustness of HASolGNN against the size of the dataset. Finally, we conduct ablation studies to test the necessity of key design components within HASolGNN.

Exp-1: Solubility Prediction Errors. First, we report the performance of HASolGNN across all three datasets compared to ten GNNs baselines in Table. 1. We note that AttentiveFP achieves the best results across all three datasets among the baselines. Compared to the best-performed baseline AttentiveFP, HASolGNN reduces the test MAE by 15.81%, 11.35%, and 29.59% on the Exp-DB, MolMerger, and BigSolDB datasets, respectively. Fig. 5 visualizes the solubility prediction errors of HASolGNN compared to AttentiveFP on the test dataset of Exp-DB. The results indicate that HASolGNN predictions align more closely with the ideal fit (where predicted values equal actual values). Specifically, compared to AttentiveFP, HASolGNN (1) achieves a lower test MAE and a higher test R-squared; (2) produces 29.92% (178 compared to 254) fewer predictions with absolute errors exceeding two (outside the two green lines).

Table 1: Comparison of Solubility Prediction Errors (Test MAE) for HASolGNN and All Baselines across Three Datasets (best results in Bold, second-Best in Italics).

Methods	Exp-DB	MolMerger	BigSolDB
GCN [28]	0.9911	0.9450	1.2926
GAT [59]	0.9538	0.8856	1.3035
GraphSAGE [22]	0.9971	0.9446	1.2851
GIN [66]	0.9519	0.9520	1.2691
GatedGNN [32]	0.9191	0.8518	1.2592
ResGatedGNN [58]	0.9606	0.8754	1.2830
CGCN [64]	1.0184	0.9758	1.3534
GraphTransformer [53]	0.9452	0.9006	1.2737
MFGNN [17]	0.9542	0.8425	1.2695
AttentiveFP [2]	*0.8688*	*0.8036*	*1.0521*
HASolGNN (ours)	**0.7315**	**0.7124**	**0.7408**
Improvements upon AttentiveFP	15.81%	11.35%	29.59%



(a) AttentiveFP. **(b) HASolGNN.**

Figure 5: Visualization of Solubility Prediction Errors of HASolGNN vs. AttentiveFP on Exp-DB (MAE & R-squared).

Exp-2: How LLMs improve on "small" datasets? To study how the LLMs module potentially improves the performance over small dataset, we randomly sample subsets that is between 1% to 10% size of the Exp-DB. For each sample size, we randomly sample 50 different sample sets and calculate the average and standard deviation from them as shown in Fig. 6. We follow the same training, validation, and test split proportion described in Sec. 5.1. We observe the followings from Fig. 6: (1) HASolGNN-LLMs consistently achieves lower errors compared to HASolGNN as the sample size grows from 1% to 10% of Exp-DB; (2) the error reduction achieved by HASolGNN-LLMs gradually reduces as the sample size increases; and (3) for both HASolGNN and HASolGNN-LLMs, the errors decrease as the sample size grows. The shrinking reduction of errors by HASolGNN-LLMs can be attributed to the higher data availability and quality as more samples are included, which enhances the effectiveness of supervised graph learning. Notably, at the 10% sampling rate, the performance gains (reduction in MAE) achieved by HASolGNN-LLMs is only 0.0114, suggesting that the benefits of the LLMs module gradually diminishes. Nonetheless, performance at small data sizes is extremely valuable given the significant cost of obtaining effective experimental data on new classes of systems.

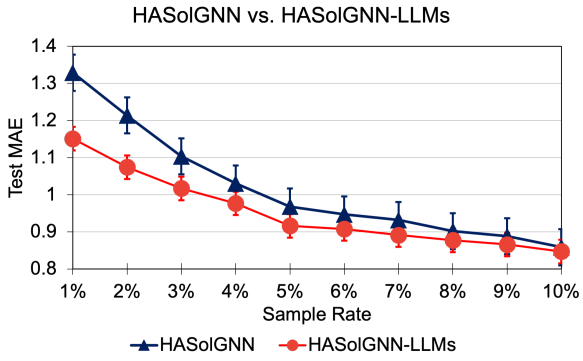


Figure 6: Performance of HASolGNN vs. HASolGNN-LLMs in Small Dataset (1%-10% randomly sampled from Exp-DB, each sampling rate corresponding to 50 differently samples).

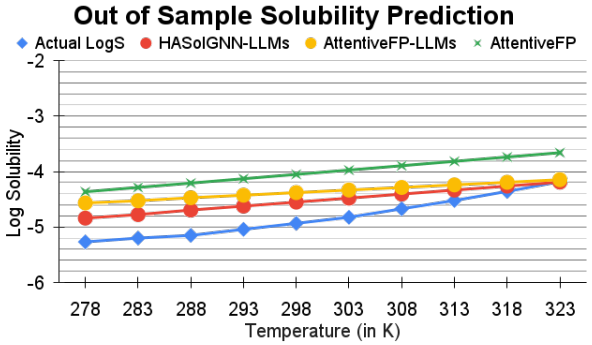


Figure 7: An Example of Prediction of the Solubility of an out-of-the-sample Solute-Solvent Pair with Varying Temperature (note: from inductive learning setting on BigSolDB, i.e.: data split based on unique solute-solvent pairs).

We next investigate how incorporating the LLMs module may help under the inductive learning setting on BigSolDB. This scenario presents challenges due to dataset sparsity, which arises from the limited amount of unique solute-solvent pairs in BigSolDB. Fig. 7 illustrates the performance of HASolGNN-LLMs and AttentiveFP-LLM in predicting solubility of an out-of-sample test solute-solvent pair across varying temperatures. We observe that incorporating LLMs module to both AttentiveFP and HASolGNN can significantly improve the solubility prediction in the inductive setting.

Exp-3: Robustness of HASolGNN. We evaluate the robustness of HASolGNN against the state-of-the-art method AttentiveFP by randomly sampling the subsets of the Exp-DB dataset, ranging from 20% to 100% of the size of original Exp-DB dataset. As the size of datasets increase, we observe two major trends from Fig. 8 that (1) the test MAE of both HASolGNN and AttentiveFP decreases slightly and (2) HASolGNN consistently outperforms AttentiveFP by maintaining comparable performance gains across all sampling rates. The first trend tells us the larger datasets enhance the performance as they provide enriched signals. In addition, the performance gains by both HASolGNN and AttentiveFP are relatively modest between 20% and 100%. Even at the 20% sample rate, the dataset contains a relatively large number of solute-solvent pairs given the large size of Exp-DB. The latter trend demonstrates that HASolGNN consistently delivers robust performance gains over AttentiveFP.

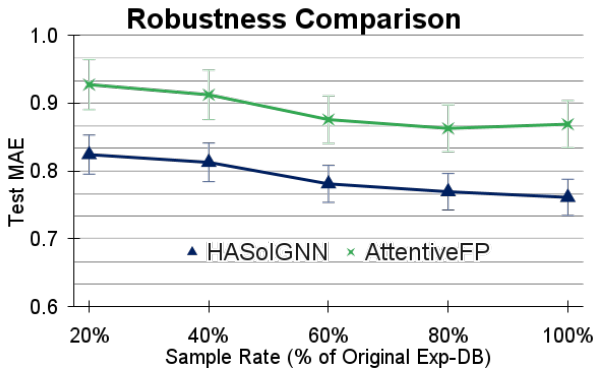


Figure 8: The Robustness of HASolGNN vs. AttentiveFP in terms of Dataset Size (sampled from Exp-DB).

Exp-4: Ablation Studies. We conduct four sets of ablation studies to evaluate the effectiveness of the key components in HASolGNN: (1) w/o IE Block: removing the Interaction-Graph Embedding (IE) Block from HASolGNN (in this case, MF_{IG} is formed by concatenating MF_{solute} and $MF_{solvent}$), (2) w/o ME Block: The Molecule Embedding (ME) Blocks are removed, leaving only one Atom Embedding (AE) Block in both MFGMs (please refer to the HASolGNN Framework in Fig. 2); (3) w/o AE Block: removing the AE Block from HASolGNN (only one ME Block left in both MFGMs, please refer to Fig. 2); and (4) w/o Sum Pooling: replacing all the sum pooling in HASolGNN by either Average Pooling or Maximal Pooling.

As illustrated in Table. 2, we observe the followings: (1) incorporating the IE Block into HASolGNN reduces the test MAE by an average of 17.16% across all three datasets; (2) adding the ME Block improves test MAE by an average of 10.24% across all datasets; (3) incorporating AE Block into HASolGNN has less impacts on prediction errors compared to IE and ME Block, reducing the test MAE by 4.72%; and (4) replacing the sum pooling with average pooling increases the test MAE by 23.93% while substituting it with maximal pooling results in an 11.02% increase in test MAE. Our experiments have confirmed the effectiveness of the key components and justify the design choices including the IE Block, ME Block, AE Block, and sum pooling. Together, these components contribute significantly to the improved solubility prediction achieved by HASolGNN.

6 Conclusion

We have proposed HASolGNN, a novel graph learning framework designed to effectively capture hierarchical, multi-level interactions, and patterns spanning atoms, bonds, both intra- and inter-molecular relationships. HASolGNN leverages a three-level hierarchical attention mechanism integrated into the AE Block, the ME Block, and the IE Block. Experimental results demonstrate that HASolGNN outperforms all the baseline models, including the state-of-the-art AttentiveFP, establishing a new benchmark in solubility prediction performance. Besides, we propose HASolGNN-LLMs which integrates a pluggable LLMs module to tackle the small dataset challenges often encountered by the scientific community. Our experimental studies have verified that that HASolGNN-LLMs yields substantial improvements in solubility prediction for small datasets,

Table 2: Ablation Studies w/o IE, ME, AE, or Sum Pooling.

I: Interaction Embedding (IE)	Exp-DB	MolMerger	BigSolDB
HASolGNN w. IE	0.7315	0.7124	0.7408
HASolGNN w/o. IE	0.8394	0.8014	1.0219
Improvements	12.86%	11.11%	27.51%
II: Molecular Embedding (ME)	Exp-DB	MolMerger	BigSolDB
HASolGNN w. ME	0.7315	0.7124	0.7408
HASolGNN w/o. ME	0.7560	0.8354	0.8491
Improvements	3.24%	14.73%	12.75%
III: Atom Embedding (AE)	Exp-DB	MolMerger	BigSolDB
HASolGNN w. AE	0.7315	0.7124	0.7408
HASolGNN w/o. AE	0.7610	0.7720	0.7601
Improvements	3.89%	7.72%	2.54%
IV: Different Pooling	Exp-DB	MolMerger	BigSolDB
HASolGNN w. Average Pooling	0.9033	0.7865	1.0214
HASolGNN w. Maximal Pooling	0.8570	0.7643	0.8045
Improvements	17.17%	7.29%	8.60%

and remains robust for out-of-sample test pairs. This suggests HASolGNN a general recipe for solubility prediction and other molecular property prediction tasks.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Waqar Ahmad, Hilal Tayara, and Kil To Chong. 2023. Attention-based graph neural network for molecular solubility prediction. *ACS omega* 8, 3 (2023), 3236–3244.

[3] Waqar Ahmad, Hilal Tayara, HyunJoo Shim, and Kil To Chong. 2024. SolPredictor: predicting solubility with residual gated graph neural network. *International Journal of Molecular Sciences* 25, 2 (2024), 715.

[4] Hamid Reza Amedi, Alireza Baghban, and Mohammad Ali Ahmadi. 2016. Evolving machine learning models to predict hydrogen sulfide solubility in the presence of various ionic liquids. *Journal of Molecular Liquids* 216 (2016), 411–422.

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[6] Maciej Besta and Torsten Hoefler. 2024. Parallel and distributed graph neural networks: An in-depth concurrency analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[7] Dixit V Bhalani, Bhingaradiya Nutan, Avinash Kumar, and Arvind K Singh Chandel. 2022. Bioavailability enhancement techniques for poorly aqueous soluble drugs and therapeutics. *Biomedicines* 10, 9 (2022), 2055.

[8] Samuel Boobier, David RJ Hose, A John Blacker, and Bao N Nguyen. 2020. Machine learning with physicochemical relationships: solubility prediction in organic solvents and water. *Nature communications* 11, 1 (2020), 5753.

[9] Mohamed Bouzidi, Faiza Yahia, Sabri Ouni, Naim Bel Haj Mohamed, Abdullah S Alshammari, Ziaul R Khan, Mansour Mohamed, Odeh AO Alshammari, Abdalla Abdelwahab, Adrián Bonilla-Petriciolet, et al. 2024. New insights of the adsorption and photodegradation of reactive black 5 dye using water-soluble semi-conductor nanocrystals: mechanism interpretation and statistical physics modeling. *Optical Materials* (2024), 116575.

[10] Jianwen Chen, Shuangjia Zheng, Huiying Zhao, and Yuedong Yang. 2021. Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map. *Journal of cheminformatics* 13 (2021), 1–10.

[11] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. 2020. Can graph neural networks count substructures? *Advances in neural information processing systems* 33 (2020), 10383–10395.

[12] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. 2017. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling* 57, 8 (2017), 1757–1772.

[13] Bhanuranjan Das, Anurag TK Baidya, Alen T Mathew, Ashok Kumar Yadav, and Rajnish Kumar. 2022. Structural modification aimed for improving solubility of lead compounds in early phase drug discovery. *Bioorganic & Medicinal Chemistry* 56 (2022), 116614.

[14] Karl De Jesus, Rene Rodriguez, DL Baek, RV Fox, Srinath Pashikanti, and Kavita Sharma. 2021. Extraction of lanthanides and actinides present in spent nuclear fuel and in electronic waste. *Journal of Molecular Liquids* 336 (2021), 116006.

[15] John S Delaney. 2004. ESOL: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences* 44, 3 (2004), 1000–1005.

[16] Xing Du, Yi Li, Yuan-Ling Xia, Shi-Meng Ai, Jing Liang, Peng Sang, Xing-Lai Ji, and Shu-Qun Liu. 2016. Insights into protein–ligand interactions: mechanisms, models, and methods. *International journal of molecular sciences* 17, 2 (2016), 144.

[17] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* 28 (2015).

[18] Yangxin Fan, Raymond Wieser, Laura S. Bruckman, Roger H. French, and Yinghui Wu. 2024. Parallel-friendly Spatio-Temporal Graph Learning for Photovoltaic Degradation Analysis at Scale. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 4470–4478. doi:10.1145/3627673.3680026

[19] Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. 2018. PotentialNet for molecular property prediction. *ACS central science* 4, 11 (2018), 1520–1530.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[21] Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. 2020. ASGN: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 731–752.

[22] Dirk C Jordan, Chris Deline, Sarah R Kurtz, Gregory M Kimball, and Mike Anderson. 2017. Robust PV degradation methodology and application. *IEEE Journal of photovoltaics* 8, 2 (2017), 525–531.

[23] William L Jorgensen and Erin M Duffy. 2002. Prediction of drug solubility from structure. *Advanced drug delivery reviews* 54, 3 (2002), 355–366.

[24] Abolghasem Jouyban, Elaheh Rahimpour, and Zahra Karimzadeh. 2021. A new correlative model to simulate the solubility of drugs in mono-solvent systems at various temperatures. *Journal of Molecular Liquids* 343 (2021), 117587.

[25] Abolghasem Jouyban, Ali Shayanfar, Vahid Panahi-Azar, Jafar Soleymani, BehroozH Yousefi, WilliamE Acree Jr, and Peter York. 2011. Solubility prediction of drugs in mixed solvents using partial solubility parameters. *Journal of pharmaceutical sciences* 100, 10 (2011), 4368–4382.

[26] Ozren Jovic and Rabah Mouras. 2023. Extreme gradient boosting combined with conformal predictors for informative solubility estimation. *Molecules* 29, 1 (2023), 19.

[27] Yeonjoon Kim, Hojin Jung, Sabari Kumar, Robert S Paton, and Seonah Kim. 2024. Designing solvent systems using self-evolving solubility databases and graph neural networks. *Chemical Science* 15, 3 (2024), 923–939.

[28] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[29] Lev Krasnov, Simon Mikhaylov, Maxim Fedorov, and Sergey Sosnin. 2023. BigSolDB: Solubility Dataset of Compounds in Organic Solvents and Water in a Wide Range of Temperatures. (April 2023). doi:10.26434/chemrxiv-2023-qqslt

[30] Sumin Lee, Myeonghun Lee, Ki-Won Gyak, Sung Dug Kim, Mi-Jeong Kim, and Kyoungmin Min. 2022. Novel solubility prediction models: Molecular fingerprints and physicochemical features vs graph convolutional neural networks. *ACS omega* 7, 14 (2022), 12268–12277.

[31] Sangho Lee, Hyunwoo Park, Chihyeon Choi, Wonjoon Kim, Ki Kang Kim, Young-Kyu Han, Joohoon Kang, Chang-Jong Kang, and Youngdoo Son. 2023. Multi-order graph attention network for water solubility prediction and interpretation. *Scientific Reports* 13, 1 (2023), 957.

[32] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).

[33] Ya Jun Li, Kui Wu, Yang Li, Yang Zhang, Jing Jing Liu, and Xue Zhong Wang. 2018. Solubility in different solvents, crystal polymorph and morphology, and optimization of crystallization process of AIBN. *Journal of Chemical & Engineering Data* 63, 1 (2018), 27–38.

[34] Baodan Liu, Jing Li, Wenjin Yang, Xinglai Zhang, Xin Jiang, and Yoshio Bando. 2017. Semiconductor solid-solution nanostructures: synthesis, property tailoring, and applications. *Small* 13, 45 (2017), 1701998.

[35] Jianping Liu, Xiujuan Lei, Chunyan Ji, and Yi Pan. 2023. Fragment-pair based drug molecule solubility prediction through attention mechanism. *Frontiers in Pharmacology* 14 (2023), 1255181.

[36] P Llompart, C Minoletti, S Baybekov, D Horvath, G Marcou, and A Varnek. 2024. Will we ever be able to accurately predict solubility? *Scientific Data* 11, 1 (2024), 303.

[37] Caicheng Long, Zixin Jiang, Jingfang Shangguan, Taiping Qing, Peng Zhang, and Bo Feng. 2021. Applications of carbon dots in environmental pollution control: A review. *Chemical Engineering Journal* 406 (2021), 126848.

[38] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. 2013. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling* 53, 7 (2013), 1563–1575.

[39] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. 2018. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical science* 9, 24 (2018), 5441–5451.

[40] JL McDonagh, Tanja van Mourik, and John BO Mitchell. 2015. Predicting melting points of organic molecules: applications to aqueous solubility prediction using the general solubility equation. *Molecular informatics* 34, 11-12 (2015), 715–724.

[41] Mei Meng, Zhiqiang Wei, Zhen Li, Mingjian Jiang, and Yujie Bian. 2019. Property prediction of molecules in graph convolutional neural network expansion. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 263–266.

[42] Christian Merkwirth and Thomas Lengauer. 2005. Automatic generation of complementary descriptors with molecular graph networks. *Journal of chemical information and modeling* 45, 5 (2005), 1159–1168.

[43] Rui Miao, Yintao Yang, Yao Ma, Xin Juan, Haotian Xue, Jiliang Tang, Ying Wang, and Xin Wang. 2022. Negative samples selecting strategy for graph contrastive learning. *Information Sciences* 613 (2022), 667–681.

[44] Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. 2019. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*. 1–15.

[45] David S Palmer, Noel M O'Boyle, Robert C Glen, and John BO Mitchell. 2007. Random forest models to predict aqueous solubility. *Journal of chemical information and modeling* 47, 1 (2007), 150–158.

[46] Gihan Panapitiya, Michael Girard, Aaron Hollas, Jonathan Sepulveda, Vijayakumar Murugesan, Wei Wang, and Emily Saldanha. 2022. Evaluation of deep learning architectures for aqueous solubility prediction. *ACS omega* 7, 18 (2022), 15695–15710.

[47] Fabrício José Pontes, GF Amorim, Pedro Paulo Balestrassi, AP Paiva, and João Roberto Ferreira. 2016. Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing* 186 (2016), 22–34.

[48] Rui Qing, Shilei Hao, Eva Smorodina, David Jin, Arthur Zalevsky, and Shuguang Zhang. 2022. Protein design: From the aspect of water solubility and stability. *Chemical Reviews* 122, 18 (2022), 14085–14179.

[49] Vansh Ramani and Tarak Karmakar. 2024. Graph Neural Networks for Predicting Solubility in Diverse Solvents Using MolMerger Incorporating Solute–Solvent Interactions. *Journal of Chemical Theory and Computation* 20, 15 (2024), 6549–6558.

[50] Paul Ruelle, Catherine Rey-Mermet, Michel Buchmann, Hô Nam-Tran, Ulrich W Kesselring, and PL Huyskens. 1991. A new predictive equation for the solubility of drugs based on the thermodynamics of mobile disorder. *Pharmaceutical research* 8 (1991), 840–850.

[51] Seongok Ryu, Jaechang Lim, Seung Hwan Hong, and Woo Youn Kim. 2018. Deeply learning molecular structure-property relationships using attention-and gate-augmented graph convolutional network. *arXiv preprint arXiv:1805.10988* (2018).

[52] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[53] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509* (2020).

[54] Parisa Shiri, Veronica Lai, Tara Zepel, Daniel Griffin, Jonathan Reifman, Sean Clark, Shad Grunert, Lars PE Yunker, Sebastian Steiner, Henry Situ, et al. 2021. Automated solubility screening platform using computer vision. *Iscience* 24, 3 (2021).

[55] N Sukumar and JE Pask. 2009. Classical and enriched finite element formulations for Bloch-periodic boundary conditions. *Internat. J. Numer. Methods Engrg.* 77, 8 (2009), 1121–1138.

[56] Nathan J Szymanski, Yan Zeng, Haoyan Huo, Christopher J Bartel, Haegyeom Kim, and Gerbrand Ceder. 2021. Toward autonomous design and synthesis of novel inorganic materials. *Materials horizons* 8, 8 (2021), 2169–2198.

[57] Elena M Tosca, Roberta Bartolucci, and Paolo Magni. 2021. Application of artificial neural networks to predict the intrinsic solubility of drug-like molecules. *Pharmaceutics* 13, 7 (2021), 1101.

[58] Liudmila Ulanova, Tan Yan, Haifeng Chen, Guofei Jiang, Eamonn Keogh, and Kai Zhang. 2015. Efficient long-term degradation profiling in time series for complex physical systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2167–2176.

[59] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

[60] Xiaofeng Wang, Zhen Li, Mingjian Jiang, Shuang Wang, Shugang Zhang, and Zhiqiang Wei. 2019. Molecule property prediction based on spatial graph embedding. *Journal of chemical information and modeling* 59, 9 (2019), 3817–3828.

[61] Michael Withnall, Edvard Lindelöf, Ola Engkvist, and Hongming Chen. 2020. Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *Journal of cheminformatics* 12, 1 (2020), 1.

[62] Jialu Wu, Junmei Wang, Zhenxing Wu, Shengyu Zhang, Yafeng Deng, Yu Kang, Dongsheng Cao, Chang-Yu Hsieh, and Tingjun Hou. 2022. ALipSol: an attention-driven mixture-of-experts model for lipophilicity and solubility prediction. *Journal of Chemical Information and Modeling* 62, 23 (2022), 5975–5987.

[63] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[64] Tian Xie and Jeffrey C Grossman. 2018. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters* 120, 14 (2018), 145301.

[65] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. 2019. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry* 63, 16 (2019), 8749–8760.

[66] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[67] Xinyi Xu, Taihao Han, Jie Huang, Albert A Kruger, Aditya Kumar, and Ashutosh Goel. 2021. Machine learning enabled models to predict sulfur solubility in nuclear waste glasses. *ACS Applied Materials & Interfaces* 13, 45 (2021), 53375–53387.

[68] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. 2019. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling* 59, 8 (2019), 3370–3388.

[69] Da Zheng, Chao Ma, Minjie Wang, Jinjing Zhou, Qidong Su, Xiang Song, Quan Gan, Zheng Zhang, and George Karypis. 2020. Distdgl: distributed graph neural network training for billion-scale graphs. In *2020 IEEE/ACM 10th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*. IEEE, 36–44.

[70] Tianyuan Zheng, John BO Mitchell, and Simon Dobson. 2024. Revisiting the application of machine learning approaches in predicting aqueous solubility. *ACS omega* 9, 32 (2024), 35209–35222.

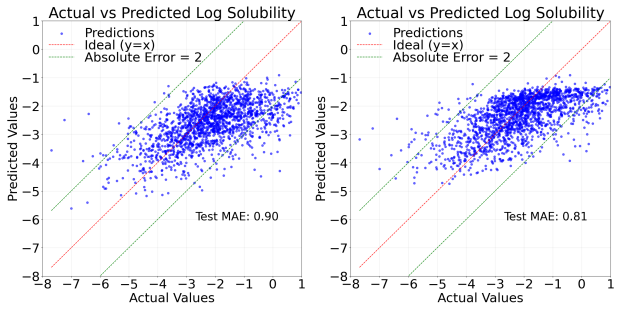
[71] Kuangqi Zhou, Yanfei Dong, Kaixin Wang, Wee Sun Lee, Bryan Hooi, Huan Xu, and Jiashi Feng. 2021. Understanding and resolving performance degradation in deep graph convolutional networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2728–2737.

7 Appendix

7.1 Appendix A: Proofs

Theorem 7.1. *The complexity of HASolGNN is* $O(ek|E|dF^2 + ek|V|F^2)$, *where* e, k, $|E|$, $|V|$, F, *and* d *denotes the number of training epochs, the number of iterations in AE Block,* $max(|E_1|, |E_2|)$, $max(|V_1|, |V_2|)$, *number of features per node, and maximum degree of solute graph* G_1 *and solvent graph* G_2.

Proof. We prove above by first demonstrating that the following two arguments hold: (1) GAT component in the AE Block asymptotically dominates over its counterparts in the ME Block and IE Block; and (2) for all embedding blocks, the GAT component asymptotically dominates over the LSTM component. For (1), AE Block operates directly on molecular graphs $\hat{G}$ while ME Block operators on synthetic graph $\hat{G}^M$ and IG operators on the interaction graph IG. Since the size of molecular graph is $|\hat{G}| = |V| + |E|$, the size of synthetic graph is $|\hat{G}^M| = 2|V| + 1$, and the size of interaction graph $|IG|$ is a constant, both $|\hat{G}^M|$ and $|IG|$ are bounded by $|\hat{G}|$. Hence, GAT component in the AE Block asymptotically dominates over its counterparts in the ME Block and IE Block. For (2), within each embedding block, each GAT is followed by a LSTM layer as illustrated in Fig. 3. Assume the embedding block operators on



(a) AttentiveFP. **(b) HASolGNN.**

Figure 9: Visualization of Solubility Prediction Errors of HASolGNN vs. AttentiveFP on MolMerger (Completely Unseen, both solute and solvent in test are unseen in training).

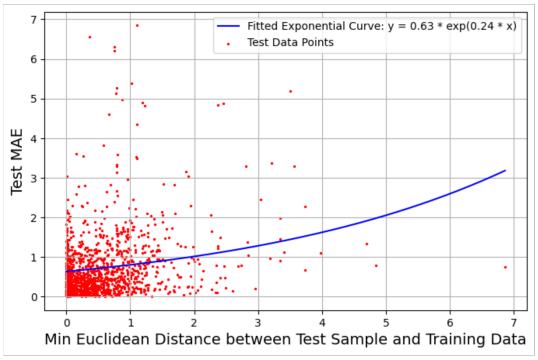


Figure 10: Similarity of Test Sample to Training vs. Test MAE.

computation graph $G = (V, E)$, each LSTM performs gates computation (four gates) which is $O(|V|F^2)$ and element wise operator is $O(|V|F)$. This cost of LSTM is $O(|V|F^2)$. The cost of a single GAT layer is $O(|E|dF^2 + |V|F^2)$ [6]. This means that within each embedding block, the GAT component asymptotically dominates over the LSTM component. Besides, one can easily verify that the complexity of the fusion mechanism is $O(F^2)$. The inference complexity of GAT is bounded by one training iteration. Therefore, the training complexity of HASolGNN is $O(ek|E|dF^2 + ek|V|F^2)$ and the inference complexity is $O(k|E|dF^2 + k|V|F^2)$.

7.2 Appendix B: Additional Experiments and Results

Exp-5: Generalization to Unseen Chemical Classes. To evaluate how HASolGNN generalizes to entirely unseen test samples, we partitioned the MolMerger dataset such that solutes and solvents in the training, validation, and test sets are mutually exclusive. This split follows a nearly 6:2:2 ratio, as detailed in Sec. 5.1. Fig. 9 visualizes the solubility prediction errors of HASolGNN compared to AttentiveFP on the completely unseen test dataset of MolMerger. This test presents the most challenging solubility prediction scenario where both solute and solvent graphs are unseen during the model training. The results demonstrate that HASolGNN predictions align more closely with the ideal fit (where predicted values exactly match actual values) in the most challenging case. Specifically, compared to AttentiveFP, HASolGNN (1) achieves a 9.93% lower test MAE and (2) reduces the number of predictions with absolute errors over two (outside the two green lines) by 35.32%.

"I have two molecules represented by SMILES strings: Solute: [CCO]; Solvent: [O]. Based on their **molecular structures**, provide specific, compact, and tailored descriptions about the **solubility** of the solute in the solvent. Discuss any **chemical properties** that influence solubility, such as **polarity**, **hydrogen bonding**, **molecular size**, or others. "

Figure 11: An Example of the Prompt Used to Query LLM.

Exp-6: Impact of Training Similarity on Prediction Errors. To investigate the relationship between training data similarity and model accuracy, we analyze how the Euclidean distance between test samples and the training set correlates with test MAE. Specifically, we measure the similarity using the concatenated molecular fingerprints of solutes MF_{solute} and solvents $MF_{solvent}$. We observe a clear trend from Fig. 10: test samples that are more similar to training data (i.e., those with smaller Euclidean distance) exhibit lower test MAE. This highlight the impact of training distribution coverage on prediction performance, emphasizing the importance of representative training data in minimizing generalization errors.

An Example of Prompt and LLM Response. We showcase an example of prompt following the template $P(S_1, S_2, Q)$ and its corresponding response from LLM. Fig. 11 illustrates an example of the prompt for querying the solubility of solute [CCO] in solvent [O]. Fig. 12 provides the corresponding response from GPT -4 for the above prompt. We highlight the key words in different colors and their corresponding responses in same color.

7.3 Appendix C: Para-HASolGNN

The design of HASolGNN is parallel-friendly [18]. We introduce a parallel algorithm for HASolGNN training, denoted as Para-HASolGNN, illustrated in Fig. 13, to scale the training of HASolGNN to large-scale solubility dataset. Para-HASolGNN exploits the following three levels of parallelism. We assume the followings: (1) the main coordinator P_0 has the access to all Level I coordinators P_i; and (2) each P_i has information access to all level II workers $P_{i,j}$.

Level I: Model Parallelism. HASolGNN contains two parallel MFGM modules. This presents opportunities for parallelizing the training by distributing solute and solvent MFGM among the processors. In each epoch, Para-HASolGNN executes model parallelism where P_i initializes parallel jobs $J_i, \forall i \in [1, 2]$. Within each the level I execution, it forward propagates $MFGM_i$ using Φ_i. At each P_i, $MFGM_i$ backpropagates independently and updates their gradients in parallel after receiving messages from P_0. The output of each module will be assembled and forwarded to $IG \in M$ by the coordinator processor P_0. P_0 calculates global loss in Eqn. 4 and updates IG.

Level II: Data Parallelism. Within the solute or solvent $MFGM$, Para-HASolGNN takes a sequence of molecular graphs as the input. At the level II, Para-HASolGNN initializes the level II parallel jobs $J_i^j, \forall j \in [1, \lceil \frac{T}{L} \rceil]$ and processes the components of the input with a

Algorithm 1 : Para-HASolGNN

Input: A batch of solute and solvent graph pairs $\Phi = \{(G_1^1, G_2^1), \ldots, (G_1^T, G_2^T)\}$, a randomly initialized HASolGNN Model M, batch size L, a coordinator P_0, a set of sub-coordinators P_i, a set of workers $P_{i,j}$, the number of epochs e;
Output: Incrementally trained M upon batch Φ.
for $m = 1$ **to** e **do**
    /* *executes model parallelism* */
    Para_Model $(J_i(MFGM_i, \Phi_i))$, $\forall i \in [1, 2]$;
    P_0.forward(IG);
    P_0 updates $IG \in M$;
return M from P_0;
procedure Para_Model(J_i)
    /* *executes data parallelism* */
    Para_Data $(J_i^j(MFGM_i^j, \Phi_i^j))$, $\forall j \in [1, \lceil \frac{T}{L} \rceil]$;
    P_i receives M_i from P_0;
    P_i updates $MFGM_i \in M$;
procedure Para_Data(J_i^j)
    /* *executes pipeline parallelism (lines 4-5)* */
    B_1.forward(Φ_i^j), $B_1 \in MFGM_{i,j}$;
    B_2.forward(Φ_i^j), $B_2 \in MFGM_{i,j}$;
    $P_{i,j}$ receives $M_{i,j}$ from P_i;
    $P_{i,j}$ updates $MFGM_{i,j}$;

Figure 13: Para-HASolGNN: Three-level Parallel Training.

"The solute is represented by the SMILES string "CCO", which corresponds to the chemical structure of ethanol. Ethanol is a two-carbon alcohol, with a hydroxyl (-OH) functional group attached to one of the carbon atoms. The solvent is represented by the SMILES string "O", which corresponds to the chemical structure of water. Water is a polar molecule with two hydrogen atoms bonded to one oxygen atom. The solubility of a solute in a solvent is largely determined by the "like dissolves like" principle, which states that polar solutes dissolve in polar solvents, and nonpolar solutes dissolve in nonpolar solvents. Both ethanol and water are polar molecules, so we would expect ethanol to be soluble in water. Ethanol is polar because of the -OH group, which has a polar bond due to the difference in electronegativity between oxygen and hydrogen. This allows ethanol to form hydrogen bonds with water, further enhancing its solubility. The two carbon atoms in ethanol contribute some nonpolar character to the molecule, but this is not enough to prevent it from dissolving in water. In terms of molecular size, ethanol is a relatively small molecule, which also favors solubility in water. Larger molecules tend to have more nonpolar character and are therefore less soluble in polar solvents. In conclusion, based on the polarity, ability to form hydrogen bonds, and small size of ethanol, we would expect it to be **highly soluble** in water."

Figure 12: Response from GPT-4 for the Prompt from Fig. 11.

fixed batch size L in parallel. The batch size is determined by available computational resources and the total workload. Besides, Para-HASolGNN optionally exploits the mini-batch data parallelism [69] to achieve even larger speed-up. It splits the information propagation of the large-scale molecular graph into parallelly computed message flow graphs induced by mutually disjoint node batches.

Level III: Pipeline Parallelism. Each MFGM comprises two ME Blocks and one AE Block. We adopt an asynchronous macro-pipeline parallelism schema [44] to parallelize the computation of the two independent branches B_1 and B_2. B_1 consists of a AE Block followed by a ME Block while B_2 comprises a single ME Block. In this way, the forward message passing of both B_1 and B_2 are parallelly computed. It eliminates the inter-pipeline synchronization (w/o information loss since the batches from level II are independent of each other).

Complexity Analysis of Para-HASolGNN. The total cost of Para-HASolGNN is $O\left(\frac{ek|E|dF^2+ek|V|F^2}{|P|} + f(\theta)\right)$. Given the input Φ and a model M, we denote the total training cost of HASolGNN using a single worker as $T(\Phi, M)$ which is $O(ek|E|dF^2 + ek|V|F^2)$. We show that for each level of parallelism, the parallel cost is in inverse proportion to the number of the workers $|P|$. We denote the communication overhead among the coordinators and workers as $f(\theta)$ which is $O(\lceil \frac{T}{L} \rceil e)$. Since L and e are hyper-parameters only relevant to the model M, $f(\theta)$ accounts for a communication overhead that is independent of the size of Φ but only dependent of the selection of parameters of M. With the level I and II parallelisms, the communication cost can be further reduced to $O(e)$. Therefore, the total cost of Para-GTrend is $O\left(\frac{ek|E|dF^2+ek|V|F^2}{|P|} + f(\theta)\right)$. $f(\theta)$ is a linear function independent of size of Φ.